User Manual

Scrivener 2.7 for Mac OS X

Literature & Latte
September, 2015
Revision: 2.7.0-01b
© 2010–2015, Literature & Latte LTD. All rights reserved.

Contents

Contents ii

Acknowledgements vi

I Introduction 1

1 Philosophy 2

2 About This Manual 5
2.1 Terms and Conventions . . . 6
2.2 Finding Things 7

3 What's New 8

4 Installation and Upgrading 22
4.1 Portable Installations 23
4.2 Registering 23
4.3 Application Updates 25
4.4 Upgrading from Scrivener 1 . 26
4.5 Staying Informed 27

5 Interface 28
5.1 Interface Overview 28
5.2 View Modes 35
5.3 Composition Mode (Full Screen Editing) 39
5.4 Selections 39

6 Quick Tour 41
6.1 Project Format 41
6.2 The Main Window 43
6.3 Outlining Your Draft 44
6.4 Scrivenings Mode 47
6.5 More About Corkboards . . . 48
6.6 Compiling the Draft 49

II Preparation 52

7 Project Management 55
7.1 Project Size Limitations . . . 56
7.2 Creating a New Project 56
7.3 Saving and Making Copies . . 57
7.4 Opening Existing Projects . . 58
7.5 Moving and Deleting Projects 58
7.6 Splitting Projects Up 59
7.7 Tips for Working Cross-Platform 59
7.8 Sharing Projects Over the Internet 60
7.9 Project Format Upgrades . . 60
7.10 Templates 61
7.11 Backing Up Your Work . . . 67
7.12 Sandboxing and Authorised Folders 72

8 Setting Up the Binder 75
8.1 How the Binder is Organised 75
8.2 Binder Icons 80
8.3 Binder Controls 81
8.4 Using Collections 83
8.5 Document Templates 90
8.6 Custom Icons 94

9 Navigating and Organising 98
9.1 General Navigation 98
9.2 Go To Menu 99
9.3 Hoisting the Binder 100
9.4 Controlling Binder Integration 102
9.5 Linking Documents Together 104
9.6 Using Favourites 113
9.7 Layouts Panel 115
9.8 Full Screen Mode 117
9.9 Filtering Items in Views . . . 118

10 Setting Up Meta-Data 120
10.1 Meta-Data Types 120
10.2 Copying Meta-Data Between Projects 127
10.3 Project Keywords 128
10.4 Project Properties 130

11 Gathering Material 131
11.1 File Import 131
11.2 Scrivener Services 138
11.3 Scratch Pad Panel 140
11.4 Print as PDF to Scrivener . . 141

12 Project Planning 142
12.1 The Corkboard 142
12.2 The Outliner 149
12.3 Project Notes 153
12.4 Setting Default Project Formatting 155

13 Cloud Integration and Sharing 156
13.1 Simplenote 156
13.2 Synchronised Folders 164
13.3 Which to Use? 174
13.4 Index Card for iOS 175
13.5 Scrivener Everywhere 180

III Writing 184

14 The Editor 187
14.1 Header Bar 187
14.2 Footer Bar 191
14.3 Viewing Media in the Editor 192
14.4 Splitting the Editor 196

15 Writing and Editing 199
15.1 Rich Text Editing Philosophy 199
15.2 Editing Basics 200
15.3 Editing with Scrivener 203
15.4 Formatting Tools 212
15.5 Inline Images 221
15.6 Using Snapshots 227
15.7 Auto-Completion 231
15.8 Text Bookmarks 233
15.9 Editing Multiple Documents 234

16 Page View 239
16.1 Page View Dimensions 239
16.2 Tips for Accuracy 239

17 Composition Mode 241
17.1 Using Background Image and Textures 242
17.2 The Control Strip 242
17.3 Floating Inspector HUD . . . 244
17.4 Customising Appearance . . 245
17.5 Composition Mode with Multiple Displays 245

18 Annotations and Footnotes 246
18.1 Inline Notation 248
18.2 Linked Notation 252
18.3 General Usage Tips for Notation 257
18.4 Text Colour and Highlights . 261
18.5 Marking Revisions 264
18.6 Text Markings and Multi-Markdown 266

19 Inspector 269
19.1 Synopsis Card 269
19.2 General Meta-Data View . . . 270
19.3 Document Support Panes . . 270
19.4 Document and Project Notes 271
19.5 References 271
19.6 Keywords Pane 273
19.7 Custom Meta-Data Pane . . . 275
19.8 Snapshots Pane 275
19.9 Linked Notes Pane 276
19.10 Locking the Inspector 277
19.11 Advanced Inspector Keyboard Usage 278

20 Scriptwriting 280
20.1 Formatting a Script in Scrivener 280
20.2 Using Page View to Estimate Page Counts 283
20.3 Importing a Script from Final Draft and Other Programs 284
20.4 Printing or Exporting a Script 286
20.5 Working with Final Draft . . 286
20.6 Working with Fountain . . . 289
20.7 Creating Your Own Script Formats 291
20.8 Using Script Formatting for Other Purposes 297

21 Writing Tools 298
21.1 Searching and Replacing . . . 298
21.2 QuickReference 309
21.3 Goals and Statistics Tracking Tools 312
21.4 Recording Audio and Photo Notes 319
21.5 The Name Generator 321
21.6 Bibliography Management . . 322
21.7 Using Equations with MathType 323

22 Using MultiMarkdown 325
22.1 What is Markdown? 325
22.2 What is MultiMarkdown? . . 326
22.3 Getting Started with MMD . 328
22.4 MMD and Scrivener 328
22.5 Compiling 334
22.6 Importing MMD Files 340
22.7 MMD Meta-Data 340
22.8 Updating MultiMarkdown . 341
22.9 Further Information 341

IV Final Phases 342

23 Creating a Table of Contents 345
23.1 How to Create a ToC 345
23.2 Contents in E-books 347

24 Compiling the Draft 348
24.1 Compile Summary 349
24.2 Built-in Presets 351
24.3 Available Compile Formats . 354
24.4 Compiling and Saving Settings 361
24.5 Managing Custom Presets . . 362
24.6 Expanded Compile Interface 364
24.7 Contents 365
24.8 Print Settings 372
24.9 Separators 373
24.10 Cover 376
24.11 Formatting 378
24.12 Title Adjustments 394
24.13 Layout 395
24.14 Script Settings 399
24.15 Transformations 401
24.16 HTML Settings 404
24.17 PDF Settings 405
24.18 Replacements 405
24.19 Statistics 409
24.20 Tables 410
24.21 Footnotes/Comments 411
24.22 Page Settings 417
24.23 Compatibility 422
24.24 Meta-Data 422
24.25 RTF Compatibility 426
24.26 Quick Font Override 427
24.27 KindleGen 428

25 Exporting 430
25.1 Drag and Drop Between Projects 430
25.2 Exporting Binder Files 430
25.3 Exporting to an Outliner with OPML 432
25.4 Exporting Meta-Data to a Spreadsheet 433

26 Printing 435
26.1 Document Printing 435
26.2 Printing Index Cards 437
26.3 Printing Outlines 439
26.4 Printing the Draft 441

V Advanced 442

27 MultiMarkdown Tips and Tricks 444
27.1 Functional Annotations and Comments 444
27.2 Multiple Footnote Streams with MMD 445

28 Customising Keyboard Shortcuts 446

VI Appendices 447

A Menus and Keyboard Shortcuts 448
A.1 Scrivener Menu 448
A.2 File Menu 450
A.3 Edit Menu 455
A.4 View Menu 457
A.5 Project Menu 468
A.6 Documents Menu 471
A.7 Format Menu 478
A.8 Window Menu 484
A.9 Help Menu 486

B Preferences 488
B.1 Preference Presets and Themes 489
B.2 General 490
B.3 Appearance 494
B.4 Corkboard 498
B.5 Compose 503
B.6 Navigation 506
B.7 Editor 510
B.8 Formatting 513
B.9 Corrections 516
B.10 Import and Export 518
B.11 Backup 522

C Placeholder Tags 524

D Tips and Tricks and Trouble 525
D.1 Getting Support 525
D.2 Using Scrivener in a Secure Environment 525
D.3 Hidden Preferences 526
D.4 Troubleshooting 532

E Project Bundle Format 534
E.1 The Files Folder 535
E.2 The Settings Folder 536
E.3 The Snapshots Folder 536

F Legal 537

Acknowledgements

Many thanks to the following people for very kindly donating time or code to Scrivener:

Heinrich Gliesen—help with inline image scaling.

Jonathon Mah—help with the bubble highlights around comments and footnotes.

Andreas Mayer—NSBezierPath and table view extensions[1].

Andrew C. Stone—smart quotes code[2].

Todd Ransom (author of Avenir) at returnself.com—filtering and page number printing code.

Jesse Grosjean (author of Mori and WriteRoom) at Hog Bay Software—auto-saving code.

Split view code based on OASplitView from the excellent OmniGroup[3].

Andy Matuschak—for various resources and great code snippets available on his website.

Matt Gemmell—various code snippets (including the colour contrast code, the use of Quick Look for the file link view and the corkboard options bubble window) from his source code site[4].

Andy Matuschak[5] and Matt Gemmell are also responsible for the basis of the code for Scrivener's HUD panels in Tiger (code to be found on their respective web sites).

Philip Dow (author of Journler[6])—help with the custom ruler code.

Wagner Truppel for help figuring out how to draw the diagonal status text in the corkboard.

Martin Wierschin of Nisus, for various help with the chicaneries of the text system and RTF.

Peter Maurer—help with the custom animated disclosure triangles (and kindly provided the images too).

Christian and Eric at Devon Technologies[7]—help with keeping Scrivener in the background when the clippings services are used.

Positive Spin Media—the tab bar in the project notepad uses PSMTabBarControl, Copyright © 2005, Positive Spin Media (all rights reserved), which in turn uses portions of code from Copyright 2004 The Shiira Project[8] (all rights reserved).

Malte Rosenau—for pointing me in the direction of the code I needed to import web pages

[1] http://www.harmless.de/cocoa.html
[2] http://www.stone.com
[3] http://www.omnigroup.com
[4] http://mattgemmell.com/source/
[5] http://andymatuschak.org
[6] http://journler.com/
[7] http://www.devon-technologies.com
[8] http://www.positivespinmedia.com
[9] http://www.bean-osx.com

with their titles intact.
James Hoover, creator of Bean[9], for providing the basis of the "show invisibles" code and for sharing various other code snippets from his work on Bean.
Brent Simmons—the OPML importer code is based on a class created by Brent Simmons for NetNewswire (Copyright © 2002, Brent Simmons).
Shortcut Recorder is © Contributors of ShortcutRecorder[10]—thanks to Jesse Grosjean for pointing me in the direction of this project.
Nathan Day—the PDF Services alias file is created using Nathan Day's NDAlias class[11].
Mark Onyschuk—help with the window flipping code for the templates panel, which is based on his LIFlipEffect[12].
Kino—the Snapshot comparison tool was inspired in large part by Kino's "Compare Documents" macro for Nisus Writer Pro.
Robert Warwick[13]—the improvements to text table support are based on code Robert wrote for Stone Hill Invoicer.
Improved PDF anti-aliasing was provided by a line of code from Skim[14].
The redoubtable Douglas Davidson, Apple engineer and guru, for responding so helpfully to so many of my enquiries about the intricacies of the Cocoa text system.
Aki Inoue at Apple for advice about the word count code.
Fletcher Penney, for contributing so much time and effort in helping me get MultiMarkdown implemented in Scrivener.
Everyone on the Cocoa-Dev list for their help and support, with special mention to Bill Cheeseman, Malcom Crawford, Max and Marcus at Blue Technologies (authors of Ulysses), J. Nozzi at Bartas Technologies (author of CopyWrite).
Stephen Kochan, author of Programming in Objective-C, for answering my questions when I was getting started.
Many thanks to the developer of Index Card, "DenVog", who went out of his way to help get syncing to work between Index Card and Scrivener.

Beta Testers

Too many to mention everyone, but a big thanks to all of you, especially those who helped beta test during the early stages - your feedback led directly to the freeware release of Scrivener Gold and a much enhanced Scrivener. Special mention to:
Alexandria Pallas-Weinbrecht, for all the help in testing Scrivener's composition mode on a dual-monitor set up.
Ernesto Salcedo, for help with a bug in viewing webarchives.
Sophie, for implementation ideas in expanding Scrivener's MultiMarkdown capabilities.

And in the best Oscars-speech-style, thanks to Kurt Vonnegut for making me want to write and my father for buying me a ZX Spectrum when I was a boy and thus forever

[10] http://code.google.com/p/shortcutrecorder/
[11] http://homepage.mac.com/nathan_day/pages/source.xml
[12] http://www.lorem.ca
[13] http://www.codehackers.net
[14] http://skim-app.sourceforge.net/

turning me into a geek.

— Keith Blount

Part I

Introduction

Chapter 1

Philosophy

> The maker of a sentence launches out into the infinite and builds a road into Chaos and old Night, and is followed by those who hear him with something of wild, creative delight.
>
> Ralph Waldo Emerson

Most word processors and text editors aimed at writers assume the creative process will take place in linear form; that is, they assume that the writer knows how his or her work will begin and will start at the beginning and continue through until reaching the end, and for those that do work that way, they assume that a linear form is a useful format for a text that spans hundreds of pages. Planning and restructuring is therefore forced into a separate workflow—the writer must either plan before beginning and keep track of hundreds of different files using the computer or face the laborious task of cutting and pasting numerous chunks of text to restructure a long piece of work at the end. For shorter pieces of writing, this is not a massive problem, but for longer texts—such as novels or academic theses—the writer can often find him- or herself battling against the tools of their trade. What a word processor does get right is in not presuming anything of your working methods. It is, at the core, ruthlessly simple.

Scrivener is part of a new breed of writing software which focusses more on providing a solid open-ended writing *tool* (as the word processor does) rather than a writing *process*, which presumes a system upon you. As with a word processor, there are no chapters or sections or scenes; how you use the software determines these things, not some feature baked into the program. Scrivener's design is fairly unique in that it easily provides a platform for writers of many philosophies and disciplines. It's a tool that works equally well for a doctoral dissertation, as it does for writing a short story or a screenplay.

Often you will hear of “pantsers” vs. “plotsers” within fiction writing circles. Those that write by the seat of their pants and those that plot everything out before they start. You will not find that this program urges you into one way of working or another. Rather you should find that its features and design provide a set of basic abilities which can amplify your method and make the mechanics of writing easier for you. If you start with a big pile of files that need to be sewn together, you’ll probably find an emergent outline to be useful. If you make the outline first, you’ll probably find the integrated outline-is-your-book approach to be useful. No longer will you need to maintain a separate outline document that needs to be kept up to date with the text of the book. The two will be one and the same. No more will you need elaborate spreadsheets to track crucial details. All of that can be woven right into the mechanical structure that is your book in this software. These principles extend to all forms of writing, from the novel to the legal brief to the documentary shot list.

Scrivener was developed because no other writing environment quite provided all of the features required for the sort of writing that demands easy access to research, a quick overview of numerous documents and the easy restructuring of swathes of text.

The main concepts of Scrivener can be summarised as follows:

- The software should allow the writer to use many of the formatting features familiar to word processors—different fonts, italics, paragraph styles and so on. It is up to the writer to avoid distractions, not the software.
- What looks good in print is not always the best way of viewing something on the screen: the software should allow the user to completely reformat the text on export or for printing without affecting the original (thus making it easy to export a manuscript in a format suitable for an editor and then export it just as easily in a format more suited to publishing on the internet).
- There should be a synopsis linked with each document that can be viewed with other synopses to get an overview of the project as a whole.
- Linked to the above, it should be easy to view the project as an outline and use drag and drop to restructure the project. Moving between the outline and the text itself should be fluid.
- It should be possible to view individual “documents” either as discrete chunks of text or combined with other (arbitrary) documents. Thus, for instance, a chapter could be viewed as a whole or each scene could be viewed individually.
- The user should be able to view more than one document at the same time—for instance, the end of one chapter alongside the beginning of the next, a character

sketch alongside a scene in which that character appears, or a research document alongside the writing it is being used to support.

- The software should be capable of storing and displaying common research documents, because writers do not just use text—they also refer to images, web pages, recorded interviews and other media.

The overriding philosophy behind Scrivener was in part inspired by a passage written by the author Hilary Mantel in a collection of essays by writers on the process of writing entitled *The Agony and the Ego*. Hilary Mantel described a process of "growing a book, rather than writing one," which can be summed up as follows:

1. During the first stage of writing, you might jot ideas down on index cards—phrases, character names, scene ideas; any insight or glimpse.
2. When you have gathered a few index cards, you might pin them to a corkboard. Other ideas build around them, and you might even write out a few paragraphs and pin them behind the index card with which they are associated. At this stage, the index cards have no particular order.
3. Eventually, you may begin to see an order emerging and re-pin the index cards accordingly.
4. After you have gathered enough material, you might take all of your index cards, sheets of paper and jottings and place them into a ring-binder. You are still free to move everything around, but now you have a good idea of how much work you have done and how much more work you have to do.

Scrivener is thus a nonlinear writing tool that provides the writer with all of the features found in any text editor along with the functionality for "growing" your work organically within the program itself in a way which will become more useful the more you write, naturally and effortlessly.

Chapter 2

About This Manual

This manual has been written using Scrivener and is available in three different formats. In addition to the PDF available from the Help menu, under User Manual, you should also find two copies which have been specially formatted to be printed on either US Letter or A4 standard printer paper on the Support section of the Literature & Latte website[1]. The Scrivener project used to author the manual can be downloaded from our website as a demonstration of writing in Scrivener. It has been written using the MultiMarkdown formatting syntax and so demonstrates that system as well.

> **Annotating the PDF**: If you wish to make notes and annotate the PDF using software such as Reader or Skim, it is recommended you download a separate copy, or locate the PDF in the Scrivener installation and drag a copy out for your own personal use. When Scrivener updates, it will very likely overwrite the existing PDF in the installation (often with revisions to the text), which will destroy any of your notes.

The manual itself has been split into five primary sections and an appendix. The first section introduces Scrivener's terminology and interface and gives a quick tour for those who want to start writing as soon as possible. It is highly recommended that you also go through the separately available Interactive Tutorial (which can be accessed via the Help menu in Scrivener) to get the most out of Scrivener and eliminate confusion over some of its unique features, but the Quick Tour (chapter 6) should have you writing using Scrivener's basic main features within half an hour.

Scrivener has a wide variety of features to accommodate many different purposes, including novels, screenplays, academic papers and dissertations, general non-fiction, journalism, blogging, and much more. While it strives to present as simple an interface as possible, once you start digging into the application, you will find a degree of flexibility and complexity to suit even the most esoteric needs. To help organise all of these concepts, the next three sections take a pragmatic look at the primary stages of any major project, Preparation (Part II), Writing (Part III), and Final Phases (Part IV), and detail Scrivener's tools for aiding you through the drafting process and ultimately exporting and printing

[1] http://www.literatureandlatte.com/support.php

your work. In each section the features most useful to you during those phases will be explained in depth. If you want to find a specific feature, consult the appendices at the end of the manual for a cross-referenced list to every menu item and window.

Users who are already familiar with prior versions of Scrivener will want to consult the What's New section (chapter 3), where new major features are briefly explained and a list of minor changes and features is provided as well.

2.1 Terms and Conventions

Some features apply only to the standard version of the software, and others only apply to the Mac App Store version. Various key features will be marked to indicate this. If you have purchased the program directly from our web site, then you have the standard version. If you used Apple's App Store tool to buy Scrivener, then you have the Mac App Store version. Sections applicable to only one or the other will be indicated as such:

- Standard retail version of Scrivener: [**Standard Version**]
- Mac App Store version of Scrivener: [**App Store Version**]

Whenever the documentation refers to an action that must be taken by you, the visible name for that action will be formatted **like so**. Button labels, menu items, and keyboard shortcuts will all be displayed in this fashion.

Menus will be displayed in a hierarchy using the "►" character to separate top-level, sub-menu, and items. Example: To convert a range of selected text to uppercase, invoke the **Format ► Convert ► To Uppercase** command.

Keyboard shortcuts will use the following symbols:

- ⌘–: The Command key, or the Apple key, is the one located directly to the left and right of your spacebar.
- ⌥: The Option key is also labelled the Alt key on some keyboards, depending on which country you purchased your Mac from. Some laptops only have one Option key on the left side.
- ^: Control is usually located to the left and right of the Option keys on their respective side. Some laptops only have one Control key between the Option key and the Fn key on the left side.
- ⇧: The Shift keys are rarely used by themselves in shortcuts but are often used in combination with other modifier keys.

When a keyboard shortcut requires a combination of modifier keys, they will be printed together. Example: ⇧⌥⌘–V, which matches **Edit ▶ Paste and Match Style**, means you should hold down all three of these modifier keys and then tap the V key on your keyboard. The hyphen is used to separate the modifier keys from the target key and is not meant to be included in the shortcut itself.

Interface elements will be capitalised as proper nouns to help differentiate them from standard nouns when appropriate. A Collection is an interface element, while the word "collection" can be used to indicate a casual grouping of items, and not necessarily a formal Collection.

Some of the names for various elements within Scrivener are customisable on a per project basis, and how you name these will impact much of the interface. A good example is the Draft, the place where your manuscript in progress is built. This can be called whatever you like, and what you name it will impact the names of menu items that refer to it. In all cases, this documentation will refer to these malleable elements by their default names.

In cases where file paths are printed, the UNIX convention of providing a shorthand to describe your personal home folder will be used. An example might look like:

```
~/Documents
```

The tilde is a shorthand which means: "/Users/yourusername". In this case, the path refers to the Documents folder in your home folder.

2.2 Finding Things

Since this PDF has been birthed within the age of digital documentation, and has yet to make its way to print form, a proper index has never been compiled for it. Despite this, in practice you should have little difficulty in locating the topic you are interested in. Modern PDF reader software features excellent searching capabilities; most things can be discovered merely by searching for the proper names of things—as labelled in menus, buttons or dialogue boxes—in the PDF.

Alternatively, the appendices have been designed to use as a sort of topical index. If you have a question about a particular menu command, for instance, you can find it in the appendix, Menus and Keyboard Shortcuts (Appendix A). Often, if the feature merits it, there will be a cross-reference to a more thorough description of the feature in the main manual. When all else fails, a detailed table of contents has been provided at the beginning of the document.

Chapter 3

What's New

Scrivener 2 is a major revision, featuring comprehensive overhauls of the interface to make it even more intuitive and easy to use. It includes a substantial list of long-awaited new features, such as a better style system, non-linear collections of documents, free-form corkboards, better snapshot integration, and much more. We understand that you are a busy person, and while new software can be fun for some, others just want to keep working in the ways they are used to. Scrivener 2 has been designed so that the majority of the new features will not get in your way until you want them, allowing you to be up and running with a minimal period of transition. When you do get the time to explore the new features, they are easily accessible and we think you'll find they add valuable tools and methods to your writing process.

In addition to the list below, you might want to also check out the new preferences (Appendix B) and menus (Appendix A).

New in 2.7

Project Format Updated

IMPORTANT: The Scrivener project format has been updated. Projects created in 2.7, or updated by it, are no longer compatible with older versions of Scrivener on either platform (for Windows, the oldest version you should use is 1.9). When you first open an older project in 2.7, you will be asked if you wish to update it and back up the current version for safe-keeping. This will primarily be of importance to those who collaborate with others, or who work from multiple Macs. Refer to Project Format Upgrades (section 7.9) for more information.

El Capitan Compatibility

Scrivener is now compatible with OS X 10.11, or "El Capitan".

A New Pair of Clothes

The main project window and numerous other panels have been given a modern look, intended to fit in better with the design of Mac OS X 10.10 "Yosemite" and beyond. This

includes a new set of toolbar icons, inspired by Apple's design in iWork toolbars, and a refreshed application icon.

Don't panic! (Most) everything is right where you left it and looks roughly like it always has. The only thing that has moved is the Inspector panel control strip, which is now along the top instead of the bottom of the sidebar.

Courier Prime

Scriptwriting mode now uses Courier Prime by default (if you have changed your script settings already, this will not impact you until you make a new script using a built-in preset or project template). We hope you find this font provides a more pleasing environment to work within while composing your scripts. When you compile to Final Draft, Courier-style font information will be stripped, so this change should not impact those using the FDX format, merely the working environment.

If you prefer working in another font, you can change the scriptwriting settings by visiting the **Format ► Scriptwriting ► Script Settings...** menu command, and change the font used by each element. Refer to Creating Your Own Script Formats (section 20.7.1) for further information on script format modification.

New in 2.6

Yosemite Compatibility

Scrivener is now compatible with OS X 10.10, or "Yosemite".

New in 2.5

Mavericks Compatibility

Scrivener is now compatible with OS X 10.9, or "Mavericks".

Localisation for French, Spanish and German

The main Scrivener interface has been translated into French, Spanish and German. This means that if you have your computer set up to prefer one of these languages, Scrivener will automatically detect that and switch to using your native language. Some prefer to stick with English because they have grown familiar with it. You will need to launch the program with the system set to English in order to do so.

Additional Changes

To see a full list of changes, visit the full release notes web page[1].

[1] http://www.literatureandlatte.com/scrivChangeList.php

New in 2.4

All Graphics Updated for Retina

All of the toolbar, interface and binder icons have been updated for Retina compatibility. You should no longer see any blurry icons when using Scrivener on one of these newer displays. The original icon sets have been retained and will be used on older equipment, so this change will be of no concern to you unless you own a Retina display capable device.

Fountain Support Added

Support for the Fountain screenplay format[2] has been added to Scrivener's list of supported import/export file formats. Fountain is a plain-text format, inspired by the likes of Markdown, and is thus suitable for writing on a wide variety of devices. Using our automatic folder synchronisation (section 13.2), you can put your screenplay in the cloud and edit from anywhere. If you wish to produce Fountain files individually, it is now possible to compile your script to the format, or export individual sections from the binder. Fountain files can also be imported into the binder as either entire scripts, or cut up automatically by scenes. Read more about Working with Fountain (section 20.6).

Integration with Scapple Files

Version 2.4 introduces integration with Literature & Latte's software, Scapple[3]. If you've been using Scapple to form new ideas and wish to bring those into Scrivener, it is easy to do so (subsection 11.1.9).

Regular Expressions (RegEx)

[**10.7+ (Lion) Only**] This advanced text search and replacement syntax can now be used in most search contexts with Scrivener. Regular expressions allow an unparalleled level of control when searching and/or replacing. If you are unfamiliar with them, but have been looking for a better way to find text in Scrivener, it is possible to find many basic tutorials on the Web. Even learning a few tricks could increase the usefulness of the search tools in Scrivener for you. At it's most advanced levels, it is a simple programming language in its own right, and thus this manual will not attempt to teach you how to use it. Read more about how they have been integrated with Scrivener, with Regular Expressions (subsection 21.1.5).

[2] http://www.fountain.io/
[3] http://www.literatureandlatte.com/scapple.php

New in 2.3

Improved Microsoft Word & OpenOffice Converters

In the past, RTF has been the primary input/output format in Scrivener, as its .doc/x and .odt export engine relied upon Apple's free utilities, which were lacking in features and had a number of formatting bugs. Scrivener how has the option of using the Aspose libraries for document conversion, resulting in .doc/x and .odt files which equal the level of quality available to RTF. This impacts both import and export/compile. Read more about the Improved Import and Export Engine (subsection 24.3.2).

Improved Layout Compile Formatting

The PDF/Print compile system has been augmented to allow for more formatting options, and now includes two separate modes, one for "publishing" and another for "proofing". The new features should allow for more direct-publishing options out of Scrivener, without having to rely upon an intermediary layout application such as word processors or desktop layout software to do so. Read more about these new options in the section on PDF output, in the compiler (subsection 24.3.3).

Locked View Modes

The particular type of view mode used to display a container or collection can be locked to that container. You could, for example, set your Character Sheets folder to always use the Corkboard view mode no matter what view mode you are preferring to use at the time in the rest of the project. Read Locking the View Mode, for more information (subsection 9.4.1).

Slide-out Sidebars

[**10.7+ (Lion) Only**] When using OS X 10.7+ with full screen mode, when the Inspector or Binder is hidden, it will automatically slide in and out of view when the mouse is moved to the far left or right of the display, or when a keyboard shortcut is used to access them. In addition, you can set an option which will automatically hide the sidebars when entering full screen, to better focus the interface on your work.

MultiMarkdown 3

The internal MultiMarkdown engine has been upgraded to the new MMD version 3.6, and its syntax generation and compiler functions have been revamped to provide support for the new features in MMD3. You can now compile directly to PDF (provided you have a functional TeXLive/pdflatex distribution installed) and ODT, including the original functions of RTF, HTML, plain MMD and LaTeX. The latter option has a new

compile option pane which makes switching document classes as easy as a single drop-down menu.

Sandboxing

[**App Store Version**] Scrivener for the Mac App Store now complies with Apple's new sandboxing rules, which means it will no longer access files on your system without express permission by you. This means that by necessity some things originally left open will now be hidden from you in the sandbox container, and it also means you may need to authorise Scrivener to access things that your projects have referred to in the past. Please read Sandboxing and Authorised Folders (section 7.12) for further details. This is mainly a concern for those who use References, linked images and aliased research documents in their projects. For most self-contained project uses, you may not notice any differences with the software at all.

Everyone, however, will want to check their preferred backup location, prior to installing the update. This is very important. If you have customised where Scrivener saves its automatically generated backups, this customisation will be lost and the new backup folder will be located inside of the sandboxed container. Open your Backup preferences panel and switch the backup location to your preferred location. This action will give Scrivener permission to access the external folder in the future.

Again, this only applies to those who have purchased the product from Apple. If you purchased Scrivener from our website or a retailer like Amazon.com, sandboxing will not apply to you, and we have no intention of making it a universal requirement. It is only applied to the App Store version as their store requires this of us.

Mountain Lion Compatible

Scrivener is now compatible with OS X 10.8, or "Mountain Lion", this includes an official Developer ID, which will allow Scrivener to run under GateKeeper. In addition to compatibility fixes, there are a few new features in 10.8 that we take advantage of:

- You can easily share your daily session goal via Twitter using the new icon in the Project Targets panel.
- If Growl isn't installed, target notifications will use Notification Center.
- Quick Look gestures (three-finger taps by default) on the binder, corkboard, outliner, synopsis search results or references lists bring up a QuickReference or Quick Look panel for the tapped item.

New in 2.2

Record Audio and Camera Snapshots

Audio notes and pictures can be taken (on Macs with an iSight camera) and automatically organised in the binder. Audio notes can be made while you work, using a recorder window which can be set off to the side and paused and resumed at will. Read more (section 21.4).

Custom Meta-Data Revamped

Until now, custom meta-data has had its home in a small little panel, hidden behind the general meta-data which shows you the status, label, and other basic information. Many people didn't even know there is more stuff behind that panel. Custom meta-data now has its own dedicated inspector pane, with easy to read & edit text fields displayed in a linear stack, rather than a small grid. Custom meta-data fields can also be shown as columns in the outliner. If this all sounds interesting and new to you, read more about Custom Meta-Data (subsection 10.1.6).

Default Subdocument Types

Assign one of your document templates to any container in the binder. New items created within that container will be set to use that template instead of the default empty file. Make easy to-do lists, quickly generate character profiles, or fill out reference tables for research you are gathering. Read more (subsection 8.5.5).

Additional Changes

Here are some new tweaks to existing tools that you might find helpful:

- Linked comments and footnotes can now be moved by dragging them from the inspector into the text editor. Where you drop is where they will be referenced from. You can still use the existing method of pre-selecting a range of text and right-clicking the note to move it.

- Session goals have been expanded to include more options. There is a night owl feature that resets after a natural day, rather than strictly going by the clock. You can also have Scrivener act like it used to, and reset whenever you close the project, or just opt to never automatically reset the session goal and handle it yourself.

- The custom icon menu has been re-organised. The last ten icons you've used will now be saved at the top of the menu for convenience. Icons that are similar in appearance or function can now be grouped together into sets.

- For workflows that require the final document to consist of one long table, such as some documentary scripts, there are now options to stitch together adjacent tables when compiling. This way you can take advantage of Scrivener's powerful outlining and section features while still producing the single table document you need.

- Keeping Scrivener Links up to date in spite of section title revisions has been a manual task. Now you can select broad ranges of text and update all Scrivener Links within that range so that their text is updated with the binder titles of the items they individually point to.

As usual, there are too many changes to list here. To see a full list, visit the full release notes web page[4].

New in 2.1

MathType Integration

Those who need to work with equations will be pleased to find that Scrivener 2.1 now offers integration with MathType, the premier equation editing and typesetting software by Design Science[5]. Adding equations is now as simple as using **Edit ► Insert ► MathType Equation**. Once added to your draft, you can double-click at any time on the equation to open it for editing.

Compile Summary Mode

The Compile feature has always been one of the richest areas of Scrivener 2.1, and also carries with it some of the most complex parts of the program. In the past, if you wanted to compile you either had to be happy with one of the pre-built formats, or dive into the full advanced interface. The new Compile Summary (section 24.1) expands the basic compile view, adding crucial and commonly needed options. Even if you already know how to use the compiler, you may find Summary mode to be of greater convenience for once involved tasks, like changing the font of a preset from one style to another.

Quick Find by Synopsis

Quickly searching for items in the Binder is now easier with the Find Synopsis (subsection 21.1.4) tool. This window features a search engine which gathers results as you type, and quick access to the index card or graphic for any of these results. Use this tool to

[4] http://www.literatureandlatte.com/scrivChangeList.php
[5] http://www.dessci.com/en/products/mathtype/

quickly find items by name or synopsis text, jot down notes in the synopsis text, or as a tool to locate and load items, without disturbing your current working area.

Data Export

Two new export features have been added to better allow exporting to spreadsheet applications and outliners, using common standards for doing so. For spreadsheets, you can now export any Outliner view to CSV (you can choose between comma, tab, or semicolon separators). This can, if you wish, be a full output of every possible column, or just the columns you have visible in the Outliner. For applications based on structural hierarchies, like dedicated outliners, you can now export to OPML.

Colour Themes

In the past you've been able to save out your preferences to a file, and so by using that you could save different colour sets as "themes". However this had a drawback in that it saved *all* of your preferences in that file, not just the appearance settings. Now you can save just appearance settings using the same **Manage...** menu from the Preference window. Themes let you switch colour sets with ease, so you could for example use a brighter Composition theme during the day, and switch to a darker theme at night.

Composition Mode

Well all right, this isn't a new feature, but we thought you should know that what we've been calling "Full Screen" all this time is now called "Composition Mode". Why? In short, Mac OS X Lion happened along, and as you may know, the phrase "full screen" is destined to become a phrase that most Mac users associate with a particular feature: expanding the application to fit the display—not completely changing the interface into something else. So Scrivener now has *two* different features for maximisation. Full Screen works like most other programs in Mac OS X Lion (and won't be available in older version of the OS), while Composition Mode is the same distraction-free writing interface you've grown familiar with.

Full Screen for Lion

[**10.7+ (Lion) Only**] For those who have upgraded to Mac OS X Lion (10.7), the new full screen feature has been adopted by Scrivener, giving you another alternative working mode where the standard project window can take up the entire screen. This is in addition to the familiar full screen editing mode, which has been renamed to Composition Mode. There are two ways to access this new feature: using the menu command

(View ► Enter Full Screen) or window button along the top[6], or by using a saved Layout that has been created in full screen mode. In the latter method, you can create special full screen layouts (such as nothing but a huge corkboard) and easily access them with the Layouts feature. To promote this, the Layouts toolbar button has been added to the default set, and the Layouts menu itself now occupies its own place in the Window menu, rather than being buried in the View menu as before.

New in 2.0

Freeform Corkboards

The corkboard feature has always been another way to view and rearrange the order of your book by placing it into a familiar metaphor. Scrivener 2 takes the metaphor the rest of the way. Now you can view a freeform corkboard for your sections where cards can be moved around arbitrarily, allowing the right brain to visualise your book in new ways. If you decide to commit the new order into the binder outline, you can do so with the press of a button. For more information, read The Corkboard (section 12.1).

Collections

Collections add tabs to your binder, giving you the ability to collect groups of documents together without moving them from their original position. It's a bit like making groups of aliases, without all of the fuss. You could create a tab to collect all of the sections your editor has marked as being too verbose, then remove them from the collection as you trim them down. Create another tab to store all of the documents that still need citations. Build smart collections that automatically look for search terms every time you view the tab. For more information, read Using Collections (section 8.4).

Marginalia

Now you get a choice between using in-line annotations and footnotes or placing your notes into a new panel located in the inspector. This not only lets you place long notes out of the way of the primary draft, it also works as a bookmarking system. Clicking on the notes in the inspector column will automatically scroll your editor to its linked position in the document. This is especially useful in Scrivenings view, where you can see

[6] The new conventional shortcut for this feature across many applications is ^⌘–F, which Scrivener now uses to enter and exit Lion's full screen mode. Previously, this was used by Project Search (subsection 21.1.2), which has had its shortcut changed to ^⌥F.

The new conventional shortcut for this feature across many applications is ^⌘–F, which Scrivener now uses to enter and exit Lion's full screen mode. Previously, this was used by Project Search (subsection 21.1.2), which has had its shortcut changed to ^⌥F.

all of the notes for the collected documents at once. Your old projects will upgrade using the in-line system as they always have, but you can easily convert to the new system with a single menu command, or vice-versa. For more information, read Annotations and Footnotes (chapter 18).

Improved Compile

For most Scrivener authors, compile is the last crucial step where all of your hard work is bound together into a single document. It lets you write in a way that is comfortable for you, while quickly generating a format that is comfortable for your editor or agent. Scrivener 2 recognises the importance of this process and has increased its abilities in nearly every facet. Now you can design export instructions which automatically title and number your chapters for you; insert custom separators between types of documents; filter compile according to meta-data or other criteria; and export to more formats, including ePub and WebArchive, with even better Final Draft support. Compile now takes your export format into account and reduces the complexity of the options provided accordingly. For more information, read Compiling the Draft (chapter 24).

iPad Workflow Support

While the iPhone and iPod touch gave people a taste of mobile writing, the iPad has made this more practical. In response, 2.0 features integration with the popular Simplenote and Index Card for iPad, as well as the ability to synchronise parts of your project text with an external folder full of files. In combination with a file-sharing service like Dropbox[7], this latter feature provides integration with many plain-text editors on the iPad and can also be used as a collaboration tool. To read more about these new features, see Cloud Integration and Sharing (chapter 13).

Better View Integration

The relationship between corkboard, outliner, and editor was, for many, a stumbling block when learning Scrivener. A lot of thought has gone into making this relationship more intuitive. You no longer have to worry about choosing between Edit Scrivenings (now called Scrivenings) and other view modes or setting preferences to change the behaviour of clicking on items in the binder. Whatever group view mode you used last is what Scrivener will use the next time, and switching among the views is now more seamless. The toolbar buttons for selecting modes have been refined to make it clearer that you are changing the way you view a binder item and/or its children items.

[7] Dropbox is a free tool that lets you keep a folder synchronised between all of your computers. For more information, visit the Dropbox homepage[8]

QuickReference Windows

Since there are occasions when it is fruitful to have more than two documents open at once, Scrivener now provides a QuickReference panel. It's a bit like Finder's Quick Look, except unlike Quick Look, you can actually edit the content and have multiple panels open at once. You can even split the QuickReference panel and edit nearly anything in the second pane that would ordinarily require the inspector to be open. QuickReference panels can also be used while working in composition mode. For more information on how to use QuickReference panels, read QuickReference (section 21.2).

Snapshots and Changes

Taking snapshots of your documents as you work has always been a crucial safety net in Scrivener, but Scrivener 1 just gave you a list of snapshots in a window and made you do the work of figuring out what had changed and which snapshot was which. Snapshots have now been moved into the inspector, meaning you can view them right alongside your current version. They can even be loaded into the editor itself as a read-only reference. You can also now ask Scrivener to mark changes between revisions, and it will highlight new words and overstrike deleted words for you. For more information, read Snapshots Pane (section 19.8).

Automatic Backups

To keep your work safe, you can now tell Scrivener to automatically back up your projects as you work. There are a number of ways in which to do so: opening and closing the project, and manual saves. Tell Scrivener to use your Dropbox folder for automatic backups, and never worry about keeping all of your machines up to date. You can still make manual backups using the File Menu as before.

Document Templates

Before, if you wrote a template character sheet, reference page, and so on, you had to make sure to remember to duplicate the original and then move the copy to the right spot. Now you can set up a template folder which will automatically place its contained documents in an easy-to-use menu from which they can be inserted into your draft wherever you want them. Document templates will be clearly marked with a special icon in the binder. For more information, read Document Templates (section 8.5).

Page View

Display your work in a variety of visual page formats, including single-page, two-page side by side, and custom page sizes. While not intended to replace your word processor or layout program, Page Layout preview will give you a rough idea of how many pages

your sections are. This will be of considerable use to scriptwriters and appreciated by others who prefer to work in a page-based method or just prefer the aesthetic appeal of seeing virtual pages scrolling by.

3.0.1 Hundreds of Other Improvements

Often when a new product version says hundreds of new improvements, you know they are counting each and every change, including bugs fixes. Scrivener 2 actually does have hundreds of improvements, far too many to realistically list. You'll notice most of the refinements as you go about working in the way you've grown accustomed to. Here are just a few of the new things you can do.

Binder Favourites: Set documents or folders as "favourites" and they'll get pinned to various document selection menus in the interface, such as Go To and Scrivener Links.

Text Bookmarks: Set bookmarks in the text of your document to mark places of interest for quick navigation. Bookmarks use the existing inline annotation feature, so they're easy to spot while you work and omit when you publish.

New Style System: Apple's built-in style system has been completely replaced with an improved style builder and selection mechanism. It is important to note that Scrivener styles are still a way of applying favourite formatting rules and font variations, not dynamic styles in the sense of a word processor, but the system for maintaining and creating these has become much easier.

Refined Menu Layout: Much thought has been put into storing the vast number of functions available. The new menus have been re-arranged to be more intuitive and logical.

Preferences Overhauled: More control is available, letting you personalise your copy of Scrivener even further, and the sections have been re-arranged to be more logical and accessible.

Better Keywords: Keywords are now stored centrally for each project and can be assigned colours which you can choose to display on index cards in the corkboard. This raises the visibility of keywords, making them more useful when viewing large groups of documents together.

Custom Binder Icons: Create or import your own custom icons and assign them to documents and folders in the binder.

Project Specific Formatting Preferences: In addition to the global application editing formatting preferences, you can now set each project to have special formatting defaults.

Improved Project Templates: Creating a new project is now even easier. Scrivener comes loaded with a number of useful starter templates, and it is even simpler to make your own and sort them using the new topical category system.

New Index Card Appearance: In addition to the Scrivener 1 index card styles, the new Rounded theme gives both the card and the "pins" an entirely new look.

Revision Marking: Set up revision marking "pens" that automatically colour new text according to the revision number as you type it in.

New Format Search: You can now search by a wide collection of format-based criteria. You can look for all italic or bold texts, search by revision markings or highlights, find inline annotations, and more.

Titles in Scrivenings: Display and edit the titles of documents right in the editor while working in Scrivenings.

Integrated Title and Synopsis in Outliner: The outliner no longer requires a second column to show both the Title and Synopsis; they are now displayed in a single integrated column, familiar to users of outlining tools like OmniOutliner, giving you more space for other columns.

View Enclosing Group: It's now easy to jump up in the hierarchy from where you are currently editing, using the current view mode. If working with corkboards, you can view the corkboard "above" the current one in the outline structure, for example, or if when working with a section of the draft in Scrivenings mode, you can broaden the scope of your text editor to include the texts around it.

Expanded Scratch Pad: The Scratch Pad has been modified to work better as an outboard research collection tool. You can now store any number of notes, and these will be saved as accessible RTFD files right on your disk. You can also easily transfer notes to any of your open projects.

Optimised Text Engine: Snow Leopard introduced changes in the way text is rendered; Scrivener 2 has been optimised to better work with these changes, improving performance when working with large documents.

Character Name Generator: Authors of fiction will be pleased to find that a powerful name generator has been built into the application. You can supply it with a wide range of criteria, including ethnic background; gender; forenames, surnames, or both; even alliteration. Scrivener also gives you the ability to add your own custom name lists.

Store Any Kind of File: You can now drag and store any type of file into your binder and load it in an external editor that supports that file type. Unlike references, these files will be stored in your project, turning it into a hub for all your file types.

Composition Backdrop: Now you can customise your composition experience by choosing images from your hard drive.

Improved Navigation: New keyboard shortcuts let you traverse the binder from right within the editor session. No more switching back and forth between binder and editor to select the next document.

Expanded Export Features: More of your work can now be saved when using the Export feature, including all of your document snapshots.

Multiple Project Notepads: Project notes can still be accessed in the inspector, but now you can also load them in a separate window, letting you view both your document notes and project notes at the same time. You can add new tabs to the project notes window to create multiple notepads, and these will be accessible from the inspector as well.

Format Bar: The slim new format bar collects the most common formatting tools into a single, compact space above the editor. No more hunting around in menus or messing with font palettes. The margin and tab stop ruler has been separated and simplified, allowing you to show only the tools you need.

Chapter 4

Installation and Upgrading

If you purchased Scrivener through the Mac App Store, then this chapter will not be relevant to you, as it pertains to the registration, installation, and maintenance of the standard retail version. You may safely skip to the next chapter, Interface (chapter 5). All installation and application updates will be handled by the Mac App Store application for you.

If you purchased the boxed version of Scrivener, insert the CD-ROM into your computer and wait for it to appear in the Finder. Create a new Finder window with ⌘–N and choose the Go ► Applications... menu item. Drag the Scrivener application from the CD to your Applications folder.

For online downloadable purchases, download the trial version from the Literature & Latte web site. The trial version can be unlocked at any time with the registration serial number that was sent to you when you purchased the application. When the DMG finishes downloading, double-click on it in the Finder (if it hasn't opened for you automatically), and follow the above instructions to drag the Scrivener icon to your Applications folder.

Scrivener will not be correctly installed on your system if you drag the icon from either the CD, or the DMG, straight into the Dock. It must be dragged into the Applications folder or, if you do not have administrative rights for the computer, you may install it into an Applications folder within your home folder (you can create one, if necessary). After copying the Scrivener application to your hard drive, you can then drag a copy to the Dock from the Applications folder if you wish.

If you attempt to run Scrivener from a "non-standard" location, you will get a warning message which will offer to you the chance to have it install it for you. If you intend to run Scrivener from outside of an Applications folder, then you should check the box that will inhibit this warning from appearing again in the future.

Once this is complete, you should eject the DMG or CD from your computer. Click the Scrivener icon in the Dock (if you have placed it there), or double-click the icon in the Applications folder to launch Scrivener.

4.1 Portable Installations

We do not recommend installing Scrivener on a portable drive that you routinely remove from your computer in order to keep your data portable between multiple machines. It is best to install and register Scrivener on *each* machine independently.

4.2 Registering

4.2.1 The Trial Version

You can try out all of the features of Scrivener for 30 non-consecutive days without having to pay or register. During that period, Scrivener will be **fully functional**. After 30 days of use, you will no longer be able to access Scrivener at all[1] unless you register. If you are getting close to the end of your trial and have decided to not purchase Scrivener, please skip forward to the sections on Compiling the Draft (chapter 24) and Exporting (chapter 25), for details on getting your data out of the program.

4.2.2 Purchasing Scrivener

During the trial period, whenever you launch Scrivener, you will be reminded of how many trial days you have left and given the option of buying and registering. Clicking on the **Buy Online** button will take you to the Literature & Latte online store, where you can purchase a licence for Scrivener. You can also visit the online store at any time whilst using Scrivener with the **Help ► Purchase Scrivener...** menu item.

When you buy a licence for Scrivener online, you will be e-mailed your unique serial number, which will have the following format[2]:

```
SCRVNR001-XXXX-XXXX-XXXX-XXXX-XXXX
```

Make sure that you keep this serial number, along with the exact name under which you registered Scrivener (the "Serial Number Name"), in a safe place, as you will need both to register Scrivener again in the future. Remember that you may need to re-register at a later date, too—for instance when installing Scrivener on a new computer, or re-installing after wiping your hard drive for whatever reason—so it is very important to store this information safely.

[1] Actually, Scrivener will allow you a single "grace" session after the 30 days so that you can export all of your work. This way, if you have decided not to buy Scrivener but forgot to export during the 30 days, you can still easily get your work out of the program.

[2] The number in the first block may be something other than '1', this is okay. However if your number is '0' you have an old serial number and will need to upgrade.

If you have lost your serial number, clicking **Retrieve Lost Serial...** in the registration window will take you to our vendor's self-service support site, where you can request to have the information sent to you.

4.2.3 Registering Scrivener

After purchasing a licence and receiving your unique serial number, you can register Scrivener by clicking on the **Enter License...** button in the "Scrivener is unregistered!" box that appears whenever you launch Scrivener. Alternatively, you can choose **Register...** from the application (Scrivener) menu.

In the "Serial Number Name" box of the registration window, enter the exact name under which you registered Scrivener. Enter your unique serial number in the "Serial Number" box, then click the **Register** button. You must make sure that both the user name and serial number are exactly as they appear in the registration e-mail you received, or registration will fail.

If you receive a message stating that the name or serial number is invalid:

- Check and make sure they are in the right order. The name of the software owner should be in the top field; the serial number in the second field.
- Try using copy and paste to transfer the information from the e-mail to this form and make sure that when you select the text in the e-mail, that the selection range does not extend beyond the first or last letter or number in the field.

For this reason, it is probably best to copy and paste both the user name and serial number directly from your e-mail. Be sure to select only the text and not any additional white space.

After clicking the registration button, Scrivener will attempt to authenticate your copy over the Internet. No private information will be relayed during this. If for some reason the Internet cannot be accessed from your computer, you will receive a warning message with instructions on how to activate the software. A URL will be provided which you can use on another computer to access the activation code you'll need to complete the registration process on the original computer. You can use copy and paste into the forms to make this easy.

Once Scrivener has been registered, you can begin using it immediately. There will no longer be a time limit on its use and you will no longer see the nag box at startup.

4.2.4 Use of System Contacts

When Scrivener is first run on Mac OS X 10.8 or greater, you will be asked by the operating system if it is okay for Scrivener to access your Contacts. This is a standard request

for permission that is generated by all programs that use your contact list. Scrivener only uses your list for one purpose: it will use information in your system's default card to generate your name into page headers, as well as generating cover sheets and the like when creating templates. If you would rather not grant Scrivener the ability to do this, you will need to fill out these details yourself.

> **Pro Tip**: In cases where your personal identity is rarely the name under which you publish, you might want to have Scrivener use another contact card entirely, rather than the system default. Rather than changing the default card to your pseudonym, you can simply set up a special address card entry for your writing persona, and then place the text "(Scrivener:UseMe)", somewhere in the "Note" area of the contact card you wish to use. This will be used to generate the headers and footers, as well as any contact information for coversheets and so forth in the templates.

4.3 Application Updates

When you first start using Scrivener, you will be presented with a panel which asks whether or not Scrivener should automatically check for updates. To use this feature, you must have a connection to the Internet.

> **Software vs. Your Work**: The advent of mobile operating systems for tablets and smart phones has unfortunately blurred the line between what a program is and what your data is. On these devices, deleting or re-installing an application can destroy your work.
>
> Standard operating systems don't work that way. They are considerably more robust and well designed for content creators, down to the very core of their technology and philosophy. Like all programs on your computer, Scrivener saves your work into files. You'll learn more about the specifics of this in the coming sections, but for now simply know that there is absolutely no connection between the software and your hard work. Upgrading versions, registering your trial, or switching to a beta build will have no impact on your data (save that you of course will not be able to open it so long as Scrivener is not installed; it will be accessible once the software is available again).

If you click **Check Automatically**, once a day when you launch Scrivener, if your machine is connected to the internet, Scrivener will do a quick check to see if there is a newer version available and will prompt you to update if one is found. (It is strongly recommended that you always update to the newest version available.) Choosing **Don't Check** will disable automatic checks, but you can always manually select **Check for Up-**

dates... from the application (Scrivener) menu. This option is available even if you also have Scrivener checking automatically.

You can change your mind later by setting this option in the application preferences, under the General tab (section B.2). There you can also adjust the frequency of the automatic checks, ranging from hourly to monthly.

When a check finds available updates, a window will appear outlining the details of the upgrade. It is recommended you brief yourself with the update notes, as sometimes changes in operation occur. You can view the full list of changes at any time on our web site[3].

To upgrade the application, click the Install Update button. Scrivener will download the latest version of the application for you, and when it is done downloading you will be presented with a button to restart the program. Click this, and after Scrivener restarts your copy will be up to date.

4.4 Upgrading from Scrivener 1

While it is possible to run more than one version of Scrivener at once on your computer, it is advisable to first delete your copy of 1.x before installing the new version. To do this, make sure Scrivener is closed, then drag its icon from the Applications folder to the trash in your Dock before following the above installation instructions. Since the application preferences have all been updated, you will need to go through the new preference system and set up your defaults again.

The project file format has been updated considerably, and new or upgraded projects cannot be opened in Scrivener 1.x versions of the application. The first time you load each of your older projects in 2.x, you will be presented with a dialogue box asking if you wish to upgrade the project format. Until you do so, it will not be possible to open these older projects in Scrivener 2.x. When updating the project, Scrivener will create a backup copy of the original project, preserved in the 1.x format, which you can choose to set aside or delete once you've verified the update was successful.

Once the update process is complete, the project will open and you can begin working immediately.

> **Updating large projects**: When updating very large, multi-gigabyte projects, be prepared to wait a while for the process to complete. A good rule of thumb is how long it would take to copy the project from one disk to another and then half that time again. If your project is so large that it will take 20 minutes to copy, then you should be prepared to wait *at least* a half hour for the project to update, as part of the update process

[3] http://www.literatureandlatte.com/scrivChangeList.php

> is creating a backup of the original. You will also want to keep this fact in mind when considering drive space. Do not attempt to update a multi-gigabyte project file on a drive that does not have enough space left on it to accommodate a duplicated copy.

In some cases, you may need to keep a copy of 1.x installed. If you are working with a colleague who has yet to upgrade to version 2, you will want to make sure not to update these collaborative projects and will need to use 1.x to interface with them. It is perfectly safe to do so, but if you elect to keep a copy of Scrivener 1.x installed, you will need to consider the following:

- .scriv files will open in Scrivener 2.x by default. To open older projects in Scrivener 1.x, you will need to drag these items onto the Scrivener 1.x icon in the Dock or use the **File ▸ Open...** menu command from within Scrivener 1.x.
- So long as you have both versions installed, the Mac might get confused over the clipping services, which can be used to clip information from other programs into Scrivener. Once you have fully uninstalled the old version of Scrivener, these clipping services should work just fine after a reboot.
- It is safe to run both copies at once, as they use different preference files.
- Scrivener's manual has been entirely moved to PDF (what you are likely reading). If you leave 1.x installed on your machine, the old Apple Help system will still have entries listed for Scrivener, so if you use the Help menu's search field (in Snow Leopard) to look up a topic, it will return search results from the wrong version's help. Instead, load the PDF in your preferred viewer and use the search tool there to find a topic.
- In some cases custom keyboard shortcuts, which are assigned using Apple's Keyboard Shortcuts tool, can get assigned incorrectly or lost.

4.5 Staying Informed

If you would like to keep up to date on the latest developments and releases of Scrivener, you can sign up for our low-volume newsletter using **Help ▸ Keep Up to Date...**. Once you submit the form, a confirmation e-mail will be sent to the address you provided. You will need to click a link within this e-mail before you will be officially added to the list. If you cannot find the confirmation e-mail after 24 hours, check your spam folders, and consider adding "literatureandlatte.com" to your white-list.

The "Keep Up to Date..." window also has links to our Twitter feed and Facebook page. We frequently publish small tricks and tips through these channels.

Chapter 5

Interface

The Scrivener interface has been carefully designed to cover a wide range of uses. At its most minimal, the interface could look no more complicated than a basic word processor like TextEdit. In this section, we will go over some of the basic interface elements that will be present in nearly every project you work in. Advanced features will be gradually introduced in their own sections as they pertain to specific areas of the writing process.

5.1 Interface Overview

When you initially run Scrivener, you will be presented with a Getting Started and project template selection window. This is actually a full-featured launching pad for all of your future and existing projects. Whenever you are ready to start a new project (the details of which you'll read about later on), such as a new novel or paper for a journal, you'll use this interface to select a starting template, or a blank starter. You can hide the "Getting Started" interface in the Options... drop-down menu in the bottom of this window.

The remainder of this section briefly summarises the main components of a typical project window. Each area will be discussed in greater detail later on, and if you are interested in a particular feature you'll find cross-references to more exhaustive articles on these topics. You may want to click on the "Blank" category in the left list, and then use the "Blank" starter to create a temporary test project to play around with, while reading this chapter. Skip forward to the Quick Tour (chapter 6) if you need instructions for doing so, but come back after you make a project, as it assumes a basic understanding of what we will cover in this chapter.

The main project interface comprises five main elements, not all of which are visible when a new project is created.

1. **Toolbar**: Frequently used tools displayed in a standard fashion, familiar to most Mac OS X applications, which can be customised as you see fit.

2. **Binder**: A hierarchal list of all items in your project, including the manuscript; a general collection area for research materials, notes, and so forth; a trash can for

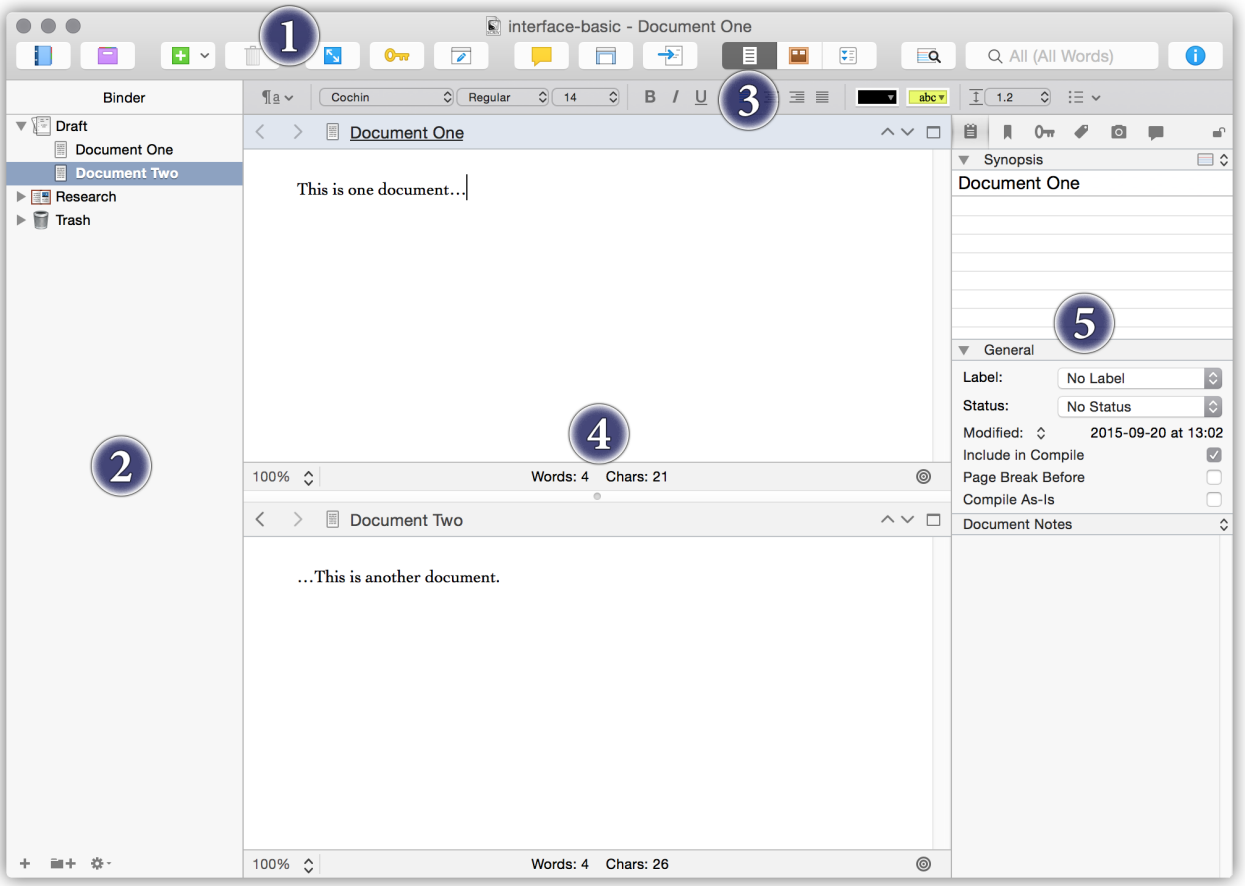

Figure 5.1: The Basic Interface

collecting deleted items; and finally the top-level area where you can create your own files and folders.

3. **Format Bar**: Supplies frequently used text editing and formatting tools, available to both editor splits.

4. **Editor**: The editor can be split into two panes which can show two parts of the same document, or two different items altogether. This is where all viewing and editing is done.

5. **Inspector**: A feature-packed tool providing information about the currently active split. The inspector has six separate sub-panes, which address various aspects of the viewed resources. Not all of them are available for all media types.

Most of the main interface sections (binder, editors, and inspector) can be resized

within the window via the grab handles located at the bottom, or in the case of splits, by dragging the divider line between them.

Each of these elements will now be explained in greater detail.

5.1.1 Toolbar

Scrivener has a customisable toolbar which includes many common functions, including a project search tool. To change the order, add, or remove icons, use the **View ► Customize Toolbar...** menu item.

It is also possible to hide the toolbar by using the **View ► Hide Toolbar** menu item. Note that when the toolbar is hidden, you can still access project search via the **Edit ► Find ►** sub-menu, or ^⌥**F**, which will provide a pop-up window featuring the same search controls available as on the toolbar.

5.1.2 The Binder

The binder is so-called because it really acts as the ring-binder for your project. It provides a hierarchical list of all items in your project. You can drag items around to restructure your project and double-click on any item to rename it. Selecting an item in the binder displays its content in the editor pane that most recently had the keyboard focus (see below) and its meta-data in the Inspector (again, see below).

You can create as many folders and files as you wish in the binder, but there will always be three top-level folders that cannot be deleted (although they can be moved around within the top level and renamed):

Draft The Draft folder is where you place all of the pieces of text you want to include in your final manuscript. As such, the Draft folder is unique in that it can only hold text files and folders—it cannot directly hold image files, PDF files, QuickTime media or web pages as other folders can, though it is of course perfectly possible to embed figures into the text itself, much as you would in a word processor, at the editor level. All of the writing you want as part of your finished work should be placed inside the Draft folder, in the order you want it (which, of course, is easy to change by dragging and dropping). When you are ready to export or print your manuscript, use **File ► Compile...**. This will combine all of the individual documents inside the Draft folder into one long document, formatted as you specify. Each file you place inside the Draft may represent a chapter, a scene, a paragraph or whatever you want.

Some templates may rename this folder to something else, but its function remains the same.

Research The Research folder can hold any type of file supported by Scrivener (text, image, media, archived web pages or PDF files). It provides a default place for non-text files to be imported and stored (although you can create other top-level folders for this purpose if you so wish).

Some templates may rename this folder to something else, but its function remains the same.

Trash When you delete an item from the binder, it is not really deleted but just moved to the Trash folder. You can drag items in the Trash folder back into the main binder, but you cannot create new items inside the Trash. To empty the Trash, use the **Project ► Empty Trash...** menu command. This will permanently delete all items inside the Trash folder (and cannot be undone).

The Collection tab interface can be toggled on and off with the associated toolbar button, or the **View ► Collections ► Show Collections** menu item. You can organise material into tabs you create, or save common searches as dynamic collection. In most cases, if this manual refers to having a "binder selection", it is often accurate to assume that applies to collections and search results as well. Collections and search results can also be dismissed by clicking the **X** button in the footer bar for the Binder. Due to the similarity of these views, the entire sidebar is often referred to in this manual as "the binder sidebar". This is meant to indicate that you needn't take the referenced action only in the Binder tab itself, but in any view that presents itself in this sidebar, such as the Search Results lists.

See Also:

- Setting Up the Binder (chapter 8)
- Navigating and Organising (chapter 9)

5.1.3 Format Bar

The Format Bar stretches across the top of the main application view, right below the toolbar, and provides easy access to the most common text formatting tools. It can be selectively hidden when not needed with the **Format ► Hide Format Bar** menu command. For more information, read The Format Bar (subsection 15.4.2).

5.1.4 Editor

The main editor occupies the large space to the right of the Binder, and will be your main working area for all manner of tasks. It automatically loads whenever you click an item in the Binder. Not only is it where you will be doing most of your writing, but it also has the ability to display groups of selected items in various useful fashions, which we

will be getting in to shortly, and is a capable multimedia viewer, too. The editor will take on different appearances and functions depending on what sort of document you are displaying and which group view mode you are in. The editor can also be split, either vertically or horizontally, giving you two independent working areas at once.

At the top of each editor is a header bar or view. This contains navigation buttons (the arrows) which allow you to navigate backwards and forwards through the document history. Next to the navigation button pair is the title of the document along with its icon. Clicking on the icon will display a pop-up menu that collects together some common menu commands. The entire header bar has three colour status modes:

- Plain grey is the default, and is what you will see unless you have split the editor.
- Blue appears when you have split editors, and indicates the *active* editor.
- A reddish appears when the editor has been locked (⌥⌘–**L**), which inhibits it from receiving binder clicks.

On the right side are two vertical arrows which allow you to navigate through the binder in a flat list fashion. You can also use the keyboard shortcuts, ⌥⌘–**UpArrow** and ⌥⌘–**DownArrow**. Using these navigation tools will also focus the binder on what you are selecting (and is one of the few instances in which an action taken in the editor will change the Binder selection).

On the far right of the header view is a button that controls the split. The icon in the button will indicate the type of split that is available - horizontal, vertical or none (close split). Option-clicking on this button will change the split type.

Clicking on this button when both editors are visible will cause the editor associated with that Header Bar to take over and the other editor to close.

Below each editor pane is a Footer Bar. This will change depending on the type of document visible and the current editor mode. When a standard text document is being viewed, for instance, it will display the word and character count along with a pop-up button for changing the text scale and a button for setting document targets. In scriptwriting mode, it will provide shortcuts for accessing various script entry macros. Various types of multimedia, and PDF, will display their own appropriate relevant information in the footer bar. Corkboard and Outliner group view modes have their own footer bar functions, too.

See Also:

- View Modes (section 5.2): for further information on the various group and text view modes available.

Figure 5.2: Inspector panel buttons, with "Keywords" selected.

- Viewing Media in the Editor (section 14.3): for details on how various read-only media are displayed in the editor.
- Header Bar (section 14.1): for more information on the informative strip along the top of the editor.
- Footer Bar (section 14.2): on how to use the strip along the bottom of the editor, which will change according to what type of document you are viewing.

5.1.5 The Inspector

The Inspector is not always visible by default in new projects, but will always be on the right side of the project window when it is visible. Its purpose is to display all meta-data associated with the document shown in the editor pane that most was recently viewed in either editor. Note that the special top-level folders in the binder—Draft, Research and Trash—have no meta-data associated with them, and therefore the inspector just shows the project notes or references when these items are selected. More on that later; for now, if the inspector appears suspiciously empty, know you have an item selected that does not support meta-data.

The inspector has six different sub-panels, which can all be accessed via buttons along the very top of the Inspector (Figure 19.1). If an associated panel has had data entered into it, an asterisk will appear in the upper-right corner of the button. The sub-panes are listed in the order they appear:

1. Synopsis, General and Notes
2. Synopsis, General and References
3. Synopsis, General and Keywords
4. Synopsis, General, and Custom Meta-data
5. Snapshots
6. Footnotes and Comments

The lock icon at the end locks the inspector so that you can reference the material in it even when switching to a different editor split.

The first three sub-panes include the synopsis card, and general meta-data in addition to their specialised information. These initial sections can be collapsed by clicking the disclosure arrow on the left side of the header bar for each. The two common meta-data sections are as follows:

Synopsis & Image The top frame shows the written synopsis for the binder item, which is what also appears in the index card representation of the item in corkboard mode, among other places. Using the top drop-down menu, you can optionally also attach an image to the item to represent it in Corkboard mode, instead of a text synopsis. Media files by default will display an image synopsis, but you can use the drop-down menu to switch to text if you prefer.

General The middle frame of the inspector shows basic meta-data for the current document. This includes the label and status (which can be renamed to something more appropriate to the current project), the modified and created date of the document (click on the arrows to switch between the modified and created date) and checkboxes that affect how (and whether) the document will appear as part of the compiled draft for exporting or printing. For more information on the types of meta-data available, read Setting Up Meta-Data (chapter 10).

The Notes, References, Custom Meta-data and Keywords panels all display the above two components. They are described as follows:

Notes Displays a notepad for the document. Each document has an auxiliary notepad associated with it; it only appears while you are viewing that item. Clicking the header bar of the notes area will bring up a pop-up menu from which you can select “Document Notes” or “Project Notes”, the latter of which is global to the project and accessible from any part of the project. If you have multiple project notepads, you may see more than one option here in parenthesis. For more information, read Document and Project Notes (section 19.4).

References The References table lets you to hold links to related material within the project itself. Links can also point elsewhere on your hard drive or on the Internet. Click the header bar to switch between document references and global project references. Use the latter to make links available throughout the project. For more information, read References (section 19.5).

Keywords Displays a list of keywords associated with the selected document. Keywords can be used to tag your document. Use the + and - buttons to add and remove keywords, or the gear menu to access the project's central keyword list. Read more about them in Keywords Pane (section 19.6) and Using Keywords (subsection 10.1.4).

Custom Meta-Data Each custom meta-data field will be displayed in a linear list with editable text fields. Fields can be added and managed using the gear menu. The fields can also be added to the Outliner as columns. Read more about Custom Meta-Data (subsection 10.1.6) and the Custom Meta-Data Pane (section 19.7).

The last two sub-panes, Snapshots and Comments & Footnotes, take up the entire inspector.

Snapshots An integrated view which displays all of the document's snapshots (a way of saving the current text of the document as a version, which can be reviewed or recalled at a later time) and the contents of the selected snapshot in the text area below. Snapshots can be created (from the state of the current text) or deleted with the + and - buttons in the header bar. The Compare button will show changes between the selected snapshot and the current text. The Roll Back button lets you revert the main editor to an older version of the text. For more information on using this, see Snapshots Pane (section 19.8), and Using Snapshots (section 15.6) for more general information on Snapshots.

Comments & Footnotes Words and phrases in the text editor can have notes attached to them. The contents of the notes will be stored in this pane. New comments can be created with the + button and new footnotes with the +fn button. Selected notes can be deleted with the - button. For more information, see Linked Notes Pane (section 19.9), and Annotations and Footnotes (chapter 18).

5.2 View Modes

There are three primary "group view modes" available in Scrivener. View modes give you a powerful way to view and modify multiple documents at once in a variety of fashions, each tailored to different tasks. View modes are generally only triggered when the editor is asked to view more than one document at a time. This can happen if you Cmd-click on more than one binder item or even if you just click on a folder. It is also possible to manually enable a view mode, allowing you to easily flesh out a single document with new index cards beneath it.

For the most part, Scrivener will anticipate what you are trying to do and react accordingly. If you click on a single text file, it will display the contents of that text file in the editor. If you then click on the group *containing* that file, it will automatically switch to your preferred group view mode, the one which you used last (corkboard by default).

An exception to this rule is when the view mode has been explicitly set to single text file mode. If you intentionally disable the corkboard to edit a folder's text, then the view mode will remain text-only until changed, no matter what type of item you click on in the binder.

A key principle is that view modes are just that. They are alternative ways of viewing a binder item and thus, like the binder, are methods for organisation. You can think of view modes as being like an elaborate extension of the binder, and in that sense they work closely with it and display information in the same top-down fashion. You can drag items from a corkboard into the binder to move those cards, or vice versa.

All file or folder item in the binder can *both* contain text and children. This is a departure from how most applications work, where the only way to organise items into other items is to use a folder. Scrivener doesn't restrict you to that, and indeed even allows you to convert a text file to a folder and back again. Consequently, even though it selects the most appropriate view mode depending upon what you click on, you can always override the default. For folders, you can turn off the current view mode and read or add text to it. For files you can select a view mode manually and then add items to them as if they were folders. When performing the latter, you will most likely be presented with an empty corkboard or outliner at first. This is fine, because you can still go ahead and add items to that view, and in doing so you will add children automatically to that file. This is a great way of expanding scenes and fleshing out rough ideas, so take a moment to play around with the view mode buttons in the toolbar, or their corresponding shortcut keys, to see how you can get the most out of Scrivener's unique file and folder structure.

The default view mode will prefer whatever mode you have selected last. If you choose outliner, then this is the mode that all containers will default to when clicked upon. It will remain this way until you manually change modes again.

The section on Project Planning (chapter 12) goes into greater detail on how to use corkboard and outliner modes to their maximum power. The section on Editing Multiple Documents (section 15.9) will cover the unique Scrivenings mode in detail.

5.2.1 The Group Mode Toolbar Control

Included in the default application toolbar is a so-called "segmented control". It is in the middle of the window, and contains three buttons embedded in a single row (Figure 5.3).

The figure depicts the control when a group of items or a container has been selected. When a single item has been selected, the label will change to "View Mode", and the icon

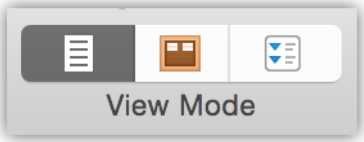

Figure 5.3: Segmented control in group mode with Scrivenings mode selected.

on the far left will depict a single page.

The control displays the active view mode with a shaded background. This is the last view mode that you picked, and it means that editor will always use that method to display groups of content until you change it again while viewing a group. To change a mode, simply click on any of the other buttons in the control. The view for the active editor will immediately change, and simultaneously set your preference for that view mode for that project.

You can also click on the shaded view mode that is already selected to intentionally disable it and view the text of the *container* that you are viewing. If you have clicked on a folder and are viewing its index cards on a corkboard, clicking the corkboard button again will switch to viewing the text of the folder. This is considered to be a view mode setting as well (or rather the absence of a preference). All future clicks to container items will load the text editor in single file mode until you again choose a view mode from the control.

These three options can also be accessed via the **View** menu, or by using the associated shortcut keys that can be found there. The shortcut keys and menus work in the same fashion, where invoking corkboard while you are already viewing an item as a corkboard will switch that view mode off and go to single text mode.

5.2.2 Corkboard

Scrivener has made the corkboard metaphor popular in modern writing software. The concept of representing many documents as index cards, letting you move them around to restructure your book, or opening them up to edit their text, is quite useful—but Scrivener 2 takes this even further. Now you are no longer constrained to viewing index cards in columns and rows; you can use the new freeform mode to move cards around as freely as you might on a desk or real corkboard. Group together cards that are related, but spread throughout your draft; spread things out in a chronological order; the choice is yours. While you can choose to commit your freeform ordering back into your draft, you can also feel free to just leave things as they are, as an alternate way of viewing a section of your work.

The important thing to realise with the corkboard is that each of the displayed index cards represent files or folders in your binder outline. They display the title and the synopsis (a short description of the item), and optionally can represent a few kinds of meta-data using real-world metaphors as well. Just remember, cards are documents, and conversely documents can all be viewed as cards.

The corkboard view is rich with features, to read more about it and how to best use it, see The Corkboard (section 12.1).

5.2.3 Outliner

While the binder is in a sense a very simple outliner, there are often times when you need to view your book, or parts of it, in a much larger context and with a greater degree of information at your disposal. The outliner view lets you see your book structure in a standard outline format with indenting used to indicate depth. It also has the ability to display many details of your documents, such as label colour, word counts, modification dates, keywords, your own custom meta-data and much more.

Since the outliner provides feedback for so much meta-data, it is a great mode for easily making and viewing bulk changes to your documents. Many people like to brainstorm in the outliner view as well, as it shows more than one level of depth at once. Whether you prefer the analogue experience of the corkboard or the more literal but expanded information in the outliner is up to you.

As with the corkboard, there are many features and uses for the outliner. To read more about it, see The Outliner (section 12.2).

5.2.4 Scrivenings

This is a unique view mode in that it works not only with the viewed items themselves, but specifically with their text, and is consequently a tool for working with text (and folder) items, rather than multimedia files. All of the text content (even empty documents) will be stacked together as if on a long spool of paper, letting you read through large sections of your book at once, no matter where they are located. As you make adjustments to the text in this view, each of the corresponding documents will be updated as you work. The overall effect is as if you were working in a single long document, but in fact you are editing potentially dozens or even hundreds of files as you go.

Scrivenings (frequently referred to as a “Scrivenings session” and known to users of version 1.x as Edit Scrivenings) are temporary editing sessions. You don’t need to worry about saving them, or what will happen if you click on something and they go away. It is merely a way of pulling together a number of files so you can edit their text at once, and then releasing them when you move on.

For more information on how to best take advantage of this editing mode, see Editing Multiple Documents (section 15.9). We will also explore a simple usage of it in the following Quick Tour section.

5.3 Composition Mode (Full Screen Editing)

Composition Mode (formerly known as "Full Screen") is a special interface which will blot out not only the rest of Scrivener, but the rest of your computer as well, allowing you to concentrate fully on the production of text. It only works for documents and folders, and while it provides access to the menus, QuickReference panels, and most aspects of the inspector, its default state is to simply display the text contents of the selected document (or documents, if you are viewing a combined text) in a "page" in the middle of your screen. Read more about Composition Mode (chapter 17).

5.4 Selections

Selection and "Focus" are often important in Scrivener, as there are features which work when only certain types of things are selected in the interface. A good example of this is the **Format ▸ Convert ▸ To Uppercase** menu command. It has no meaning when a selection of index cards is currently active, rather than some text. If there is a selection restriction on a feature, this manual will often indicate as such using the following terminology:

- **Active**: this relates to "focus", or where the application is currently accepting keyboard input. In other words, if you tapped the 'h' key, that is where the 'h' would be sent. In some cases this might not do much or anything at all; in an active editor it would insert an 'h' letter, and in the Binder it would select an item starting with 'h'.

- **Active selection**: expanding on the above, an active selection is when something is *selected* within the active area of the application. An example might be some selected text in the editor, or three selected items in the binder or corkboard. Active selections are often used as the target for various commands in the menus. An example would be the **Documents ▸ Convert ▸ Formatting to Default Text Style**, which would reset the text formatting for any selected documents in the active selection.

- **Inactive selection**: A selection can be inactive. If you select the Draft folder in your Binder and then click in the editor, the selection will become dimmer (or grey, depending on your settings). Inactive selections are seldom used, but there are a few cases, especially when exporting.

- **Active text editor**: the main text editor must be active (the insertion caret must be blinking within it or text must be selected). In cases where the interface is split into two editors, this not only means any editor must be selected, but the editor you wish to perform the function in. Some commands do not require text to be selected, only an editor, others will perform universal actions if no text is selected, but an editor is active; others will refuse to work unless you have a selection.

Chapter 6

Quick Tour

Before going over this section, it is suggested that you skim through the prior chapter, introducing the interface, if you haven't done so already.

This chapter will get you up and running with a simple sample project and will introduce a few of the most basic concepts you'll need in order to start working in thirty minutes or less. Scrivener is a complex program with a lot of potential depth, but we hope this section will demonstrate that despite this, it is also very straightforward and intuitive, so its complexity can be gradually learned as required. There is often a temptation to dive in and learn as much as possible about a new tool, but if you are here for the same reason as most people who've picked up this software, it is because you are serious about writing, and so the most important thing this manual can do is get you doing that as quickly as possible, in a fashion which is not too dissimilar from what you may already be used to.

We will go over the following tasks:

- Creating a new project from a blank template
- Navigating the binder and adding new documents to the Draft folder
- Working with the corkboard and Scrivenings text view
- Compiling the document into an RTF file

If you need to learn more than the above, we strongly suggest going through the full tutorial, which can be found in the Help menu.

6.1 Project Format

Scrivener is a project based application, which means that it stores your projects as separate files, rather than working like a database which hides the location from you and loads everything automatically when you start up the program, or a file based program like Word, which creates an empty document whenever you launch it. The intended use of a project is to store everything relating to a single major work, be it your next novel or

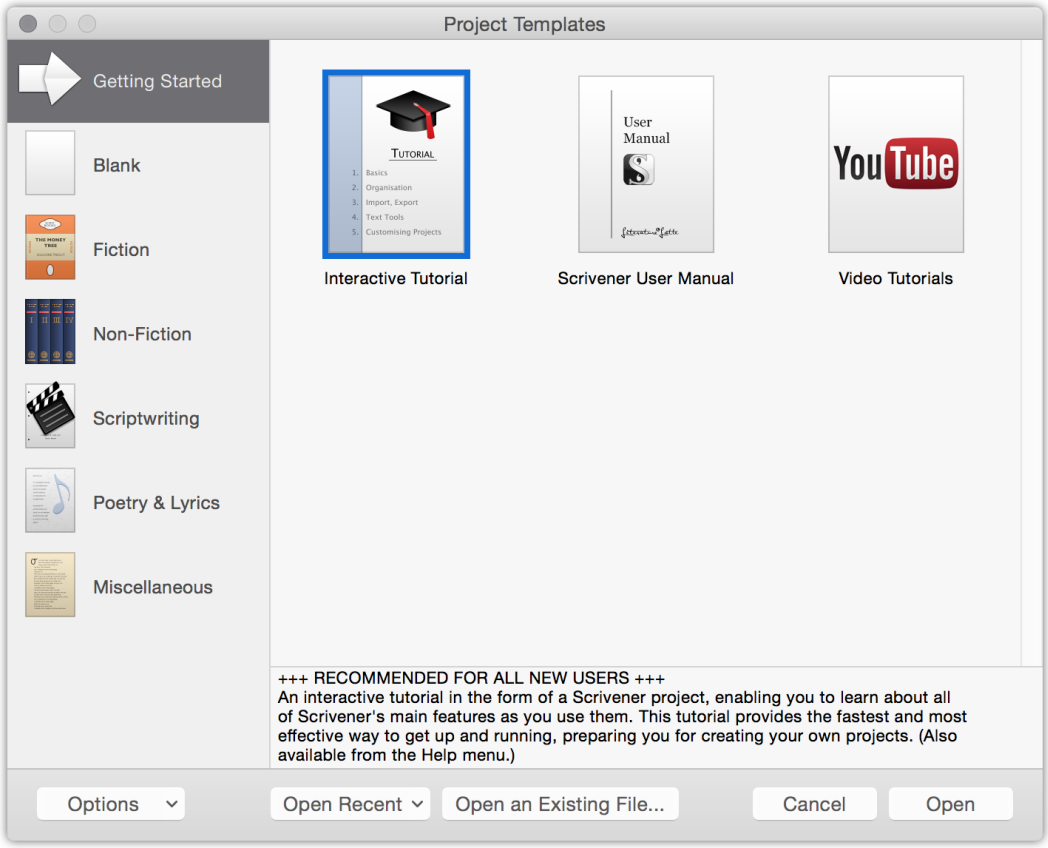
Project Templates
Getting Started
Blank
Fiction
Non-Fiction
Scriptwriting
Poetry & Lyrics
Miscellaneous
Interactive Tutorial
Scrivener User Manual
Video Tutorials
+++ RECOMMENDED FOR ALL NEW USERS +++
An interactive tutorial in the form of a Scrivener project, enabling you to learn about all of Scrivener's main features as you use them. This tutorial provides the fastest and most effective way to get up and running, preparing you for creating your own projects. (Also available from the Help menu.)
Options
Open Recent
Open an Existing File...
Cancel
Open

Figure 6.1: Template Wizard

screenplay, a dissertation for your masters, or a serial collection of articles for a magazine or blog.

It is recommended that for each new "real world" project you embark on, a new project file should be created[1]. When you create a new project, Scrivener asks where you want to save the file. By default this location is your personal Documents folder, but you can choose to organise your projects on the computer however you like. It is a good idea to choose a place you will remember. Even though Scrivener provides convenience functions for finding the projects you've worked on most recently, and will even reopen your last session for you if you choose, these tools should not be relied upon as your sole reference point for where projects exist.

Okay, enough theory. Let's create a new project right now. Invoke the **File ▸ New**

[1]There are, of course, many exceptions to this. A series of articles for a magazine might work very well together in a single project.

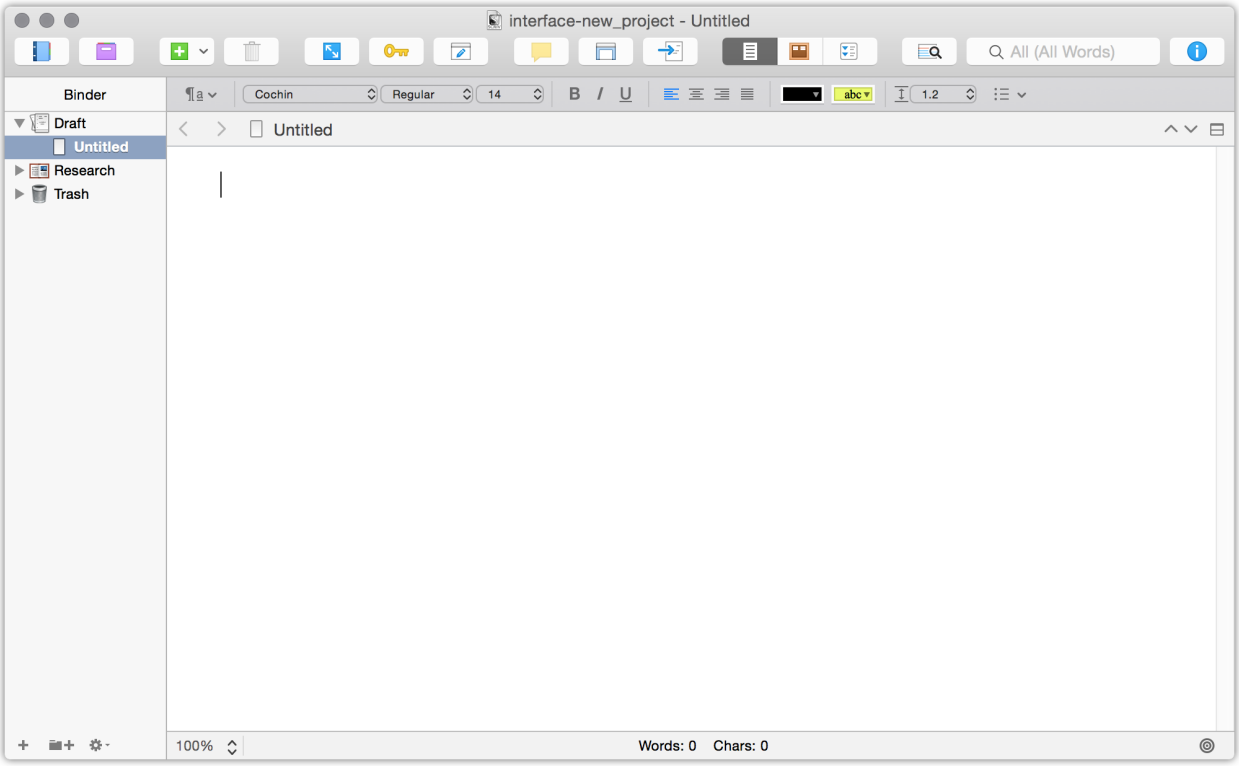

Figure 6.2: A typical new blank project

Project... menu command (⇧⌘–**N**) if you do not already see the template selection interface (Figure 6.1). In the left list of categories, click on the "Blank" entry (if necessary), and then select the blank starter project. Click the **Choose...** button in the bottom right, or just press **Return**. A window will appear asking where you wish to save the file. Choose a location you will remember (the default is your Documents folder, but you might wish to create a special sub-folder just for Scrivener projects), and give the project a name in the "Save As" field. This name is what will be used on your disk, and so should be memorable and placed in a location you'll easily find in the future. Click the **Create** button once you've decided on a file name and location.

6.2 The Main Window

The project will be created, and you'll be presented with a fresh Scrivener project window (Figure 6.2).

The left sidebar panel is your project binder, where everything stored or created in your project will be represented in an outline structure. On the right side of the window

we have the editor (which is more than just a text editor, as we will shortly see). In most cases, what you click on in the binder will be displayed on this side of the window. Along the top, above the editor, is the Format Bar, which provides access to common formatting tools such as font selection, justification, line spacing, and so on. If you cannot see all of the tools, try expanding your window horizontally. If you want to see tab stops and margins, use the menu command **Format ▸ Show Ruler** or press **⌘–R**.

Directly below the Format Bar is the editor Header Bar. The title of the current document is printed here ("Untitled" since we haven't named it yet). There are also some buttons for moving around in your project and adjusting the editor view, but we'll get to that later. The important thing to notice right now is that the Untitled document has been selected in the binder, and that this selection is being viewed in the editor.

Below the editor text area is the Footer Bar which contains information about the editor session, such as your current word and character count, and often contains useful buttons for adjusting the view.

The cursor is blinking in the text editor, and with zero words so far, it's time to get to work!

6.3 Outlining Your Draft

Why do that when we can procrastinate for another minute, though? Take a look at the binder. Let's discuss how best to use Scrivener's Draft folder. Unlike a word processor, Scrivener's design philosophy revolves around creating and organising your writing project in arbitrary pieces according to what is most comfortable for you. For example, you could make a new file for each chapter, or take that even further and make a new file for each scene *within* those chapters. It is totally up to you, and unlike in most programs, you won't be penalised for breaking things down as small as you desire. The binder displays these pieces like an indented outline. It is easy to change your mind later, so don't get hung up on which is "right", that may very well change for each project, or as you progress in your familiarity with the program.

Let's try it. If you haven't touched anything yet, the cursor should still be blinking in the main text editor (if it isn't, just click in the middle of this white text area with your mouse). Type a few words in here and then press the **Return** key to create a new paragraph.

Glance up to the top-left close window button (which is red on most Macs; or the left-most traffic light button). You should see a little dot within the button. This means your project hasn't been saved yet. If you don't see a dot, try typing another word into the editor. By the time you finish examining it, the dot ought to disappear—your project has been saved.

> **The Implications of Auto-Save**: Whenever you pause for two seconds, Scrivener's auto-save feature will kick in and save all your recent work. This is a very key thing to be aware of, because many people have cultivated the habit of relying upon their last save point as being a bit of a milestone. They can write freely and in an experimental fashion, and if what they've done displeases them they can just revert to the saved copy on the disk. The good news is that Scrivener is designed to work with this method, and in fact its way of doing so is in general superior to using the traditional model, because whenever you put down an anchor point, or a Snapshot, in Scrivener, you leave a trail of them. You can view them all in a list later on, preview their text, and roll back to any point in a document's history. This is a little out of scope for this quick tour, so for now put it on your list of things to look into if this is how you work, and when you're ready to start writing, check out Using Snapshots (section 15.6).

You might have noticed that the title of your document (printed above the editor, as well as in the binder) changed when this happened. If you leave a document as "Untitled", the first line will be used as the title (or the first few words if it is a long line, like a paragraph). This will only happen once, and you can always change the automatic title to something more descriptive if you wish (you can also turn this feature off).

Let's make a new document in the binder. There are a number of ways to do this, but first let's try the most direct method: click the large green + button in the toolbar just once.

The cursor will flip over to the binder and let you type in the name of the new document. Type something here, like "Masked Avenger", and press Enter to confirm. You'll notice that the content you just typed into the editor has disappeared. This is because you are now viewing "Masked Avenger"; you can verify this by reading the title in the editor header bar. Let's fix the name of the first document while we are in the binder. Press the UpArrow on your keyboard once. Immediately, you should see the words you just typed re-appear in the editor. That is because you are now viewing the first starter document.

To change the name, just press Esc and type in "After Math", once again pressing Enter to confirm the name change.

You might notice something going on with the icons in the binder (Figure 6.3). The document "After Math" looks like a page of text, but "Masked Avenger" just looks like a blank sheet of paper. That's because we haven't typed anything into the editor for "Masked Avenger" yet. Scrivener lets you know which parts of your project are still placeholders.

For now, let's use another method for adding new documents. When you are in the binder, there is a simple way to add new entries below the currently selected item. Select

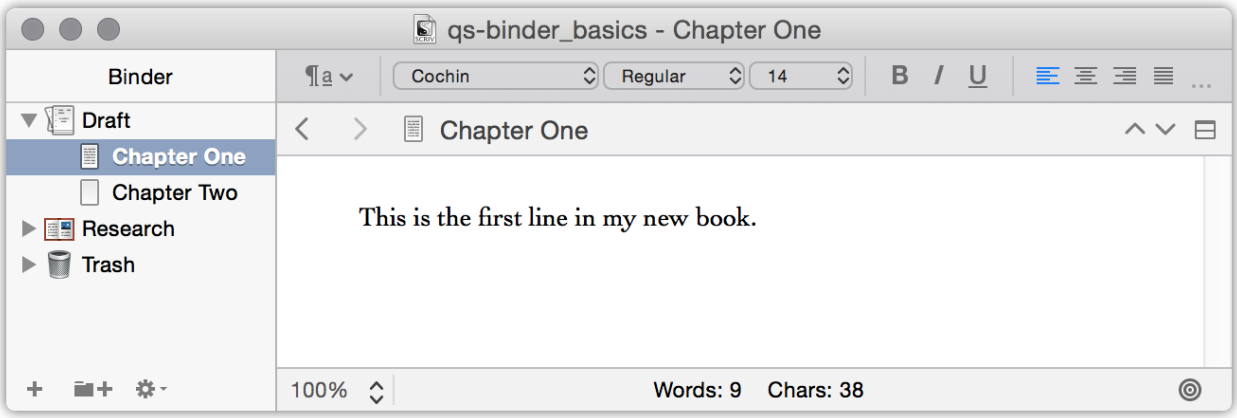

Figure 6.3: Progress so far

"Masked Avenger", and just press **Enter**. Call this one "Wineries". You can quickly build outlines this way.

Now we will take a look at a great way to get an overview of your Draft. Either click on "Draft" in the binder with your mouse, or simply press the **LeftArrow** to jump up to it. That key selects the "parent" of whatever item you currently have selected, in this case, the Draft. When you view a folder in Scrivener, by default it will automatically switch to showing you the corkboard in the editor area. You should see three index cards, one for each of the items you've created (sometimes referred to as 'Scrivenings'), but the cards will be blank; this is because they display the *synopsis* for a document, not its internal text. Double-click in the empty ruled text area on the "Wineries" card and type in a short synopsis for this section of the book. When finished, pressed **Enter** to confirm.

At this point, if you look in the binder you'll see a third icon type that looks like a small index card, next to "Wineries". This new one indicates that you've typed in a synopsis, but haven't yet typed in any text for the section. Once you enter text into the editor for the "Wineries" section, the icon will be replaced with the one you see for "After Math". Some people like to work in a top-down fashion, building a conceptual outline and then filling in what they intend for each section in the synopsis. These three levels of icons give you an overall idea of your level of progress with this task.

We'll try that now. Load the "Wineries" card by double-clicking on the icon next to the title. If you can't find the right spot, just click on it in the binder. Type in a short sentence and observe that, as predicted, the icon has now changed to match the "After Math" section.

You might be wondering how much text each section should hold. The answer is largely up to you. With a word processor, it is often convenient to work in longer doc-

uments because lots of small windows can create unmanageable clutter, but Scrivener allows you to break things down into smaller pieces, since it organises all of these files for you and keeps them in a logical order in the binder. In fact some people break things down into pieces as small as paragraphs, others stick with chapter-sized chunks. You can use index cards to represent beats, scenes, whole chapters, or even a mix of these depending on how much attention you need to give that portion of the book. It's all very flexible, and in many ways you can use the Scrivener outline to structure things according to how you think, not according to how the book will appear. That may change later on when you are ready to compile, but this can be ignored for now; there are good tools for breaking things down and merging things back together at whim. Working in a fluid fashion at every stage of your project is more important than trying to force yourself to work toward a strict goal.

6.4 Scrivenings Mode

Scrivener contains features intended to make that balance a little easier to work with. Unlike most applications of this type, you can choose to view many different sections at once and edit them as if they were a single file without actually merging them all together permanently. Doing this is easy; let's try it now. Select the Draft item in the binder again, this time by clicking on it once with the mouse. You'll get a corkboard again; now invoke **View ► Scrivenings**, or press **⌘–1**. You are now viewing the collection of documents as a Scrivenings session, rather than a Corkboard.

Each section will be divided with a horizontal rule, letting you know where one ends and the next begins. Try clicking in the blank area between the two sections you've already added text to, and keep an eye on the Header Bar: note the title changes to let you know where you are, "Masked Avenger". Add some text by typing normally, then select "Masked Avenger" in the binder all by itself. Note that your edits have been made directly to the document. It couldn't be easier.

Now we will do something else that is impossible in many programs. Right-click on "After Math" and select **Convert to Folder**. The icon will change to a blue folder with a small page in the corner. That page means the folder has text stored in it, in the same way a text file might, and indicates that the text you added to this document earlier is still there. This differs from stating that a folder has text items as children beneath it. The page badge will only be applied if the folder item itself has text content. In Scrivener, folders and text files are pretty much the same thing; it's yet another demonstration of how you can work fluidly now, and worry about details later.

Let's move the folder text into a new child document: with the "After Math" item selected in the Binder, press **⌘–N**, which is another way to create new documents. When

a folder is selected in the Binder, new documents will automatically be assigned to it as children. Call this new document, "Scene A", and then click on "After Math" again to select the folder and view it as a Scrivenings session. Scrivener always remembers the last "view mode" you used. Since we switched to viewing items as Scrivenings, the project will go on working that way until we switch it to something else (such as the default Corkboard).

The first line of text you typed will be visible above the divider line, we want to move it below the divider line, moving it into "Scene A". You can select paragraphs in Scrivener by triple-clicking anywhere within them. Try that now, and then drag and drop the text below the divider. Notice that the page icon badge in the Binder has been removed from the folder[2], and the Scene A icon now has a page of typed text in it. Once again, anything you do in a Scrivenings session is instantly made to the underlying sections it represents. You could also have used standard copy, cut, and paste commands to move the text.

No doubt you can see that Scrivener is very comfortable with the notion of splitting things up, and with this feature you can feel free to work in small portions without losing sight of the big picture.

6.5 More About Corkboards

For our next experiment, select the **View ▸ Corkboard** menu item to return to the corkboard view. What you are seeing here is an important concept in how view modes integrate together. The editor was in Scrivenings mode, giving you access to each file's text content, but is essentially focussed on the "After Math" folder. When you switched to using Corkboard mode nothing changed about what you have selected, but rather how you are viewing it. What you are looking at is the corkboard for the "After Math" folder, which currently only contains one card, "Scene A". Try clicking the Draft again in the binder. Now you are viewing the index cards that belong to the Draft. "Scene A" has disappeared because it is a child item of "After Math", not the Draft. Meanwhile, the "After Math" index card looks like a stack of cards. Folders will always look this way, even if they don't have cards within them.

- Corkboard mode will only show you one layer of depth in your outline at once, but you can tell when index cards are folders or files that contain subdocuments by the stacked appearance of their index card. Double-click on the "After Math" index card's icon and you'll return to where you were at the start of this paragraph. You've navigated "into" the "After Math" corkboard.

[2] If that didn't happen, you might have an empty line or two still in the folder text area.

- Outliner mode, which we haven't explored yet, is a bit more like the Binder in that you can see items at all levels of depth beneath the current selection. It also has the ability to show many columns of tabular data, and is great for tracking word counts per section, just to name one usage.

- Scrivenings mode usually includes all text content beneath the current selection, no matter how deep they are in the outline.

There is one main exception to these rules of thumb: Multiple Selections, which happen when you select more than one binder item at once. But, for simple single selections such as what we've been doing so far, the above guidelines will help you decide which view mode to use, depending on what you need to do.

You might be wondering about that oddly titled "Wineries" card. Let's move it to the Research section of your Binder, since in our hypothetical little universe here, it would be used to collect research on various wineries that the characters are involved with. There are multiple ways of moving items around in the outline, but the most familiar will be drag and drop. Select the Draft folder again, then click and hold on the Wineries index card in the Corkboard, and drop it onto the Research folder over in the Binder. This demonstrates how easy it is to move material in and out of the drafting area, as well as the fact that you can use a view mode to help organise your material in the Binder. In larger projects, the Binder might become quite large, so remember this trick as it can make moving items from one remote section of the Binder to another very easy. As you learn more about the program, you'll find other ways to specify content as being "meta" to the draft manuscript itself, even while leaving it in the Draft.

Hopefully by now you can see the relationship between index cards, the binder, and the text within the items in the binder a bit better. Feel free to play around with things a bit more before proceeding to the next step.

> **Working with Existing Material**: Likely, you already have content from an existing work in progress you would like to bring into Scrivener for your first project. While the methods described here could certainly help you with that, you might find the Documents ► Split ► sub-menu to be even more useful in converting a long document into an agile outline. For more information on this technique, explore Splitting The Document (section 15.3.3).

6.6 Compiling the Draft

We've only scratched the surface of what can be done, but before ending this tour we'll look at one more thing: compiling. Compile is Scrivener's terminology for what most

applications refer to as Exporting. This name was chosen to reflect the fact that Scrivener is doing quite a bit more than just converting the data from one format to another or moving data to an external location that other programs can access. Scrivener's compiler can generate documents to a number of different formats, do simple or complex transformations on the binder structure, reformat everything to a consistent look, and much more. The philosophy of letting you work the way you want to work is very much supported by the compiler. There is no need to work in Courier, Times New Roman, or whatever your editor or publisher prefers. You can choose an elegant font or one that is easy to read at 3am with sore eyes. In the end Scrivener will generate a document to your required specifications.

Let's try it now. Press ⌥⌘–**E** or select the menu command, **File ▸ Compile...**. After all of the features described above, you might think you got the wrong function at first. By default Scrivener presents a very simple form with some industry standard presets, some common options, and an export file format chooser.

- From the "Format As" menu, select "Standard Manuscript Format"
- Change the "Compile For" option to Rich Text (.rtf)

That's it. Click the **Compile** button, choose a name and location for your RTF file, and once the dialogue box dismisses, open the file using the Finder. It should open with your system's default RTF editor. As you can see, all of the test lines you typed in are there, with the header names prefixed with chapter numbering, and hash marks in between the sections as well, those came from the Compiler too, as a part of the manuscript preset. This can all be changed, or turned off entirely. Notably, the text you typed into the Wineries card will *not* be here. That's because we moved it outside of the Draft folder. The compile feature works exclusively[3] with the Draft folder.

Let's try one more thing. Close your RTF document, switch back to Scrivener and open up the compile window again. Leave everything the same, but this time click the "All Options" tab at the top. The full compile interface will appear, and with that, you'll be able to see just how much power is available. Feel free to explore the interface and do a few test compiles to see how things change. For example, if you don't like the hashmark separator between scenes, click on the Separators panel and change the first option "Text Separator" to something else. Many of the options are self-explanatory, but if you find yourself confused over something, read the relevant sections in the chapter on Compiling the Draft (chapter 24).

[3] Well, all right, not entirely exclusively, but that's a topic for another time!

There is much more to learn, but if you are in the middle of a major project and want to get started right away, you should now know enough to make a simple outline and start writing content into that outline, and then compile your work as a single draft document. As you've seen from the amount of flexibility we've demonstrated, you can rest easy on the topic of "doing things the right way from the start". Unlike many programs, Scrivener makes it easy to change your mind later on. Folders can become files, items can be easily re-arranged, merged, or split apart, and so on; so don't worry too much about adhering to any proper ways of working. Scrivener has no standard except for *your* standard, which will develop in time as you become more familiar with the software.

> **Importing Existing Work**: Chances are, you've already got some words down in another program somewhere. Scrivener supports many different text formats, and even a few data formats as well, like outliner formats (section 11.1). Importing material is often as simple as dragging and dropping the file into the project binder. After doing so, you may wish to split it up into more manageable pieces (section 15.3.3). A good rule of thumb is that RTF is the best. It is Scrivener's native file format, so if you can export your existing work to the RTF format before dropping it into the Binder, that will generally be the best solution.

If you have a spare hour or two, it would be really beneficial to go through the full interactive tutorial or watch the introductory videos[4] available on the Scrivener homepage. Together, they cover Scrivener's features in much more detail and should address most of what you should know to get the best results in your work.

Further recommended reading:

1. Interface (chapter 5)
2. Navigating and Organising (chapter 9)
3. Gathering Material (chapter 11)
4. The Editor (chapter 14) and Writing and Editing (chapter 15)
5. Compiling the Draft (chapter 24)

For now, happy writing!

[4] http://www.literatureandlatte.com/videos/index.php

Part II

Preparation

> The pages are still blank, but there is a miraculous feeling of the words being there, written in invisible ink and clamoring to become visible.
>
> Vladimir Nabokov

One of the principal goals of Scrivener is to be your first stop when a new major project is embarked upon. The program therefore contains a variety of tools which address the early phases of a writing project. These include: brainstorming, outlining, collecting and organising research, gathering prose snippets, building background and world-building information, and more.

Another Scrivener principle is that what works best for your readers and even your editor might not work best for you as a writer. As writers, we see the text in a different way, and we have different demands on how we organise that text. A table of contents is great for a reader, but is it the best tool for a writer? What if you could have your own table of contents, one that evolved out of the structure of the work itself, to a level of detail you require and no less or greater?

This section will cover the majority of the tools that you may use to accomplish these goals. If your methods couple the processes of writing and preparation, you might find it beneficial to read portions of both this section and Writing (Part III). The level of fuzziness between preparation and writing in Scrivener is very intentional, because the same tools you use to build up initial structures and ideas will be the tools you use to write, edit, and finish off your text with. In Scrivener, there is no separation between outline and book order or content. This seamless approach will help you get straight into the writing phase, even while you are still planning and evolving the work. Conversely, you can just easily start writing prose immediately, and let the bigger picture emerge organically out of what you write.

To better learn the fundamentals of Scrivener—such as where your works will be saved, and how to use the binder to select items within your project to view or edit them—it is recommended that you read the next three chapters. After that point, we'll discuss some more practical research, planning and project usage techniques, which can be learned as needed.

The topics that will be covered in this part are:

- Project Management (chapter 7): Creating, saving, backing up, and managing projects on the disk.
- Setting Up the Binder (chapter 8): Using the built-in sections of the binder, managing collections, organising your work, and setting up document templates.
- Navigating and Organising (chapter 9): Optimise how you navigate within a project.
- Setting Up Meta-Data (chapter 10): The various forms of meta-data available for tracking progress and types.
- Gathering Material (chapter 11): How to collect material from other resources and applications.
- Project Planning (chapter 12): How to rapidly capture your ideas and give them shape, shapes that will eventually become your manuscript.
- Cloud Integration and Sharing (chapter 13): How to take your work with you, collaborate with others, and manage cross-platform usage between Mac OS X and the Windows operating systems.

Chapter 7

Project Management

Scrivener is a project based application and stores your projects on your computer in the locations you designate when you create them. In this way it is much like any program that loads and saves files to your disk—rather than a database style program where all of your information is managed behind the scenes. These project files are saved into your Documents folder by default, though you can choose to organise these projects however you like, even after you've created them. Some find it convenient to rely upon the File ► Recent Projects menu item to organise their projects, or simply just let Scrivener maintain which projects are open whenever they start Scrivener for the day. These tools are valuable, but shouldn't be a substitute for good organisation on your computer, as the tools can sometimes lose track of projects.

The intended use of a project is to store everything relating to a single major work, be it your next novel, screenplay for a film, a doctoral dissertation, or a serial collection of articles for a magazine. This can be approached in a flexible manner. It is possible to use a project as a daily journal or maybe a collection of random things you intend to one day utilise in future projects. Ultimately, the choice is up to you, and there are ways to work around the default assumption where the Draft folder contains a single work. These alternatives will be explored in more detail much later on, but you should know early in the process that if you feel you would benefit greatly from having shared research and notes for multiple real-world projects, you won't be deeply penalised for doing so. Also you should know that merging and splitting projects is fairly painless, so even if you change your mind later, it won't be a major setback.

Going a bit deeper, the project file is a cohesive folder of files, which will appear as a single bundled package on a Mac, containing all of the pieces that make up your project using industry standard formats. The accessibility of this format is meant to be used as a last resort safe-guard, not as a way to allow you to edit a project in places where Scrivener is not available. **Project damage and data loss can occur from attempting to edit the internal files by hand**.

7.1 Project Size Limitations

Since Scrivener was primarily written with the long-form author in mind, much effort has been put into making the project format as robust as possible. It can handle book-length manuscripts with ease, store large quantities of research material, and handle many thousands of individual components, even on a single corkboard. Scrivener has been tested against projects with millions of words in them; way beyond what it would normally have to face. So for ordinary usage, you will never need to worry about limitations.

There is one caveat to keep in mind, however. The bigger your project is on the disk, the longer it will take to produce backups. When combined with the automated backup system, this could mean waiting long periods of time for backups to complete in the most extreme cases. While Scrivener is very capable of handling large amount of media, some users have found it better to use database software in conjunction with Scrivener, when gigabytes of data are involved.

There is no universal rule of thumb on upper limits, this will be whatever you are comfortable with, and how much available storage space you have in order to keep consistent backups. If you have an 8GB project, that means each backup will consume another 8GB maximum (less if you use the slower zip archival option), and will take as long to produce as it would to duplicate 8GB of data on your hard drive.

7.2 Creating a New Project

When you first start Scrivener, you will be presented with the Project Templates window (Figure 6.1), which also includes a handy "Getting Started" section. You may hide (or choose to reveal it again) this at any time using the **Options...** drop-down menu in the lower left of this window. In addition to providing an easy selection mechanism for new projects based on templates, you can also create a blank new document, open existing projects from your disk, or select from a short list of recently opened projects. To call up this window later on, use the **File ▶ New Project...** menu command, or press ⇧⌘–**N**.

Along the left side of the window are template categories. A number of useful templates have been provided, and it is easy to make your own as well. The **Options** drop-down menu provides template management tools. To read more about creating and managing templates, see Templates (section 7.10).

To create a new project based on an existing template, click on that template to make it active, or select the "Blank" template to start with a fresh project that can be constructed to your needs. A description of the template, if one has been provided, will appear in the grey text box below the selection area. Click the **Choose...** button to proceed. Enter the name of your project in the "Save As" field. This is what you will use to identify your project in the future, and it will also be used to name the folder (which will contain your

entire project) saved onto your disk. It can be changed later, on the disk, if you please, so if your work in progress doesn't have a name yet, don't worry!.

The next step is selecting *where* you would like to save the project file. It is a good idea to choose a place on your disk that you will remember. You might want to create a special folder in the Documents folder just for your Scrivener projects, for instance. You can use the **New Folder** button to do so.

The **Create** button will activate once you have supplied a name and valid save location. Click this button to proceed with the settings you have chosen.

7.3 Saving and Making Copies

While you work, your projects will be automatically saved on a frequent basis. By default, this means that after two seconds of inactivity, the project will always be saved. You can monitor this process by watching the upper-left corner of the window. The left-most "traffic light" button will have a dot inside of it whenever the project contains edits which have yet to be saved to the disk. If you stop typing or clicking in the interface for a moment, the mark should disappear after a brief pause, and now everything is saved to the disk. You can change how rapidly this happens by adjusting the inactivity interval in the General section of the application preferences.

At times, or if you have selected a very long interval for some reason, you might wish to manually save the project. This can be done as per normal in any application, by selecting **File ▶ Save**, or pressing ⌘–S. This feature need not be used to fully save the project, however. Very often, by the time you activate this command, the project will already be saved to the disk and it will do nothing. However if you work rapidly and rarely produces pauses, it might be a good habit to get into.

It is also possible to save a separate copy of the project, while working, and continue working in the *new* copy. To do this, select **File ▶ Save As...** (⇧⌘–S), and choose a new name for the project. After supplying a new name and clicking the **Save** button, the active project will become the new one you created, and the old copy will be closed. This can be a useful tool when you wish to experiment with a series of radical structural changes. If you merely wish to create a backup, or a trail of major milestones, but continue working in the primary project, read about creating backups on the fly (subsection 7.11.3). It is also recommended you use this method when creating copies of the project with the intention of storing or sharing it over the 'net. See Sharing Projects Over the Internet (section 7.8) for more information.

Additionally, since Scrivener projects are just files and folders on your system, you can use the Finder to manage projects, create duplicates, and archive old versions. **Always be sure to close your projects before doing so**.

7.4 Opening Existing Projects

Existing projects can be opened in a variety of ways. For convenience, Scrivener keeps track of the last several projects you have opened, and stores them in a list, accessible from **File ▶ Recent Projects...**. In addition, this list can be accessed from the Project Template window, via the **Open Recent** drop-down menu at the bottom.

By default, Scrivener will remember any projects you leave open when you quit, and will re-open these for you the next time. This behaviour can be changed in the General preferences tab.

You can open older projects using **File ▶ Open...** (⌘–O), or click the **Open an Existing Project...** button in the Project Templates window.

As with managing projects that have been saved, you can also use the Dock, Finder, Spotlight, or other system tools to open files stored on your drive. Simply double-click the project you wish to open in the Finder. Spotlight is also handy when a project gets misplaced and you are unsure of where it is saved. If you don't recall the name of the project, you can search for ".scriv" to find every Scrivener project in your Spotlight index.

When trying to sort out the differences between near similar versions of a project, you will find that the Quick Look information for your projects is quite extensive. A full outline of the entire Draft will be presented to you, including any synopses that have been typed into index cards. Use this ability to figure out which project is the correct version, before opening it.

7.5 Moving and Deleting Projects

Despite being folders full of files, you can otherwise treat the project as you would any other file or folder on your system. Everything that Scrivener needs to work with the project is contained within the project file, and so moving it will have no detrimental effect on it, and since everything is self-contained within this folder, it is a handy way to transfer all of your working data to another machine.

Always close a project prior to moving it. While a project is open, Scrivener maintains a very close connection with the files inside of it. Since it cannot predict where you will move the project, if it is moved while open the software will no longer save files in the correct location, and can lead to lost work.

Projects act just like ordinary files in the Finder. The fact that they are special folders can be largely ignored, and so they can be easily moved just as you might move a JPEG file you downloaded, or a .doc file you have written in Word.

As with all file based programs, it is not possible to delete old and unwanted projects from within Scrivener. To remove projects you no longer want, use the standard file management tools as you would with other folders and files on your computer.

7.6 Splitting Projects Up

Occasionally, one or more portions of your project might exceed the scope of your original planning and become what should be their own projects. A classic example of this might be a chapter in a biography that ends up becoming another biography about another person entirely, or a short story that keeps growing and starts to become a novel. Another common example are from those who use a dedicated Scrivener project as a commonplace book, or idea gathering tool. They might start developing an idea in this project and once it is ready to become a screenplay or what have you, they'll want to move it out of the idea journal.

Fortunately, nothing could be simpler. There are two ways of going about this:

1. Create a new project for the pieces which merit moving them out of the parent project. You can use whatever template you wish for this, or even blank, it doesn't matter. Now with the original project open alongside the new one, position the windows so that you can see both binders at once. Now simply drag and drop the pieces from the old project into the new project's binder. That's it, you're done.

2. Use **File ► Save As...** to create a fork copy of the entire project. Once you run this command, you will immediately begin working in the fork. Simply delete everything from the binder that isn't necessary for this new project.

The first method might work better if your needs are simple. If you do not have a lot of meta-data and never work with linking items together, it's safe to do. However if you use a lot of labelling, custom meta-data or status markings, the second method will be better. Most meta-data *will* transfer, even snapshots, when dragging and dropping. Anything that requires the project itself be set up a certain way however, like particular labels with predefined colours, will be more touchy and transfer, and Scrivener Links will always fail (references, so long as they point to items within the dragged selection, will work, so long as they are all moved at once).

At this point in your reading (if you are going through this in a linear fashion) some of the above might be arcane knowledge, so feel free to bookmark this page if you feel you'll need it in the future, so that once you know more about the things you can do in Scrivener, you can make a more informed decision.

7.7 Tips for Working Cross-Platform

The Scrivener project format[1] is fully compatible with its Windows companion. No conversion is necessary, and both versions can work off of the same source file (at differ-

[1] On a Mac, it is important to note that only Scrivener version 2.x and greater is cross-platform compatible. If you intend to use the Windows version as well to edit your projects, you should consider upgrading

ent times; no project should ever be opened more than once). The primary difference in appearance between the two platforms is that Windows does not have a "package" or "bundle" format like the Mac does. Thus, the Scrivener project will appear in its ordinary state, which is a folder. This is invisible to a Mac user, but in fact there is no difference between the two.

To open a project on the Mac, you need only double-click the "MyProject.scriv" project bundle, or open it from within Scrivener. On Windows this will be a folder, so you will need to descend into the "MyProject.scriv" folder and select (or double-click on) the "MyProject.scrivx" file that you will find at the top level within that folder.

To transfer projects between computers, always make sure to copy the entire "MyProject.scriv" folder from Windows, **not** just the .scrivx file by itself; the entire folder is your project. If you intend to use e-mail, file sharing services or other common methods of sharing files over the Internet, refer to the next section, Sharing Projects Over the Internet (section 7.8).

7.8 Sharing Projects Over the Internet

If you wish to share a project using the Internet, you might think to attach it as an e-mail, or upload it to a file sharing service so that one can download it or save a copy to an online backup service. Given that a project is a folder of files (whether you can see that aspect or not on the disk), the best way to send a project as a single "document" is to use zip files (or whatever archive format you prefer).

There are numerous ways to create zip files, including using the operating system itself, but if you're looking for an easy way to do so from within Scrivener, try using the File ▸ Back Up ▸ Back Up To... menu command from the project you wish to send off. Within that dialogue box you will find an option to back up as a zip file. This will "pack" your entire project into a single compressed file which will transfer across the 'net more quickly and safely.

> **Opening zipped projects**: Zip files do not load in Scrivener, you will need to "unpack" the project from the zip file before it can be opened. This is usually as simple as double-clicking on the zip file and extracting the contents to your working folders.

7.9 Project Format Upgrades

Sometimes we need to modify how Scrivener works with project, and in these cases it will be necessary for the software to update the project format. This will entail modify-

to the latest version.

ing how various internal components of the project work, and will not modifying your content, merely how the software works with it.

When opening an older project you will be asked if you wish to update the project. Click the **Update Project** button to proceed with the project update. It is safe to cancel, your project will not be touched by this version of Scrivener.

> **Coffee Break Alert**: Larger projects may take some time to update. A complete copy of the entire project will be created before the updating process is started, so for projects that measure in the gigabytes, it may take quite a while for the full process to complete. It is strongly encouraged that you let the process complete fully, rather than interrupting it. If you are in the middle of a busy schedule or coming up on a hard deadline, it might be best to defer upgrading Scrivener until you have time to devote to the process.

The first action that will be taken is to create a complete backup of the project in its current state in the same folder where the project is presently located. If you have a project called "My Thesis.scriv" in your Documents folder, then a backup project called "My Thesis Backup.scriv" will be placed in your documents folder.

Next, the main project will be updated to the new format. At the conclusion of the process, the project will open.

Trouble? If for some reason the project won't update correctly or open at the conclusion of the process, the backup created earlier will be handy to restore from. Again using our example from above, set aside or delete the "My Thesis.scriv" project file and rename "My Thesis Backup.scriv" back to how it should be, then try again. If you have repeated difficulty updating the project, please contact technical support[2].

Projects used with Scrivener 2.7 for Mac and 1.9 for Windows will not open in older versions of the software for either platform. Make sure to update all of your machines to these versions or greater, and make sure that your colleagues are up to date as well if you collaborate by sharing a project.

7.10 Templates

Simply put, project templates are ordinary projects which have had some basic structure and settings added to them. In some other applications, templates are almost a way of modifying the way the entire application behaves and so in a way the presence (or lack) of templates as *features* define the shape of what the software can do. With Scrivener, it's best to think of them as example starter projects, or even better yet, a learning tool.

[2] http://www.literatureandlatte.com/support.php

Many of them come with a few example items added to the binder, but these are meant to be helpful guides for laying out your book, not forms that you have to fill out or features that must be worked around. The items added to templates are like any other items you've added to projects on your own. They can be deleted, modified, duplicated, or set aside and ignored.

In fact, you can create your own templates for future use, which will be covered in more detail in the following pages. This is often a useful way of further customising Scrivener's default behaviour, as many settings found in menus are specific only to the projects you make them in. Since the built-in templates are just starter projects, you can use them to create your own derivative templates based upon them. Like many others, you may decide a better starting point for you is just the blank starter, or a slightly modified version of it with a few of your preferred project settings.

7.10.1 Using Built-in Templates

Most of the built-in templates will provide you with a readme-style document at the very top of the Binder that explains what the purpose of the template is, what has been modified and set up for you, and how best to use it to develop your project. Most also contain a sample PDF showing how the final manuscript will be formatted if all of the project settings are left alone.

For templates that do not come with a help file, here are a few things to keep in mind as you explore templates:

It is worth saying again: templates are just starter projects. With the exception of a few and somewhat technical differences, there is nothing in a template-generated project that is different from a blank project. In fact, a blank project is essentially an extremely minimal template. Everything that you see in a template can be handled just as you would handle any other items in the binder. They can be duplicated, deleted, modified, turned into document templates, or what have you. Scrivener templates do not provide you with "forms", "wizards", or parts of the interface that you must fill out before proceeding. A sample chapter with an empty scene in it is just what it looks like: a folder with a file in it. To change it, edit it as you would any other piece of outline. If you want another one to make a second chapter, duplicate it. Scrap it if you don't like the style at all and want to design your own structure.

When you use a new template for the first time, check the project meta-data settings (**Project ► Meta-Data Settings...**) and make sure that everything is arranged the way you prefer.

Some templates will change the name of the Draft and Research folders to something more appropriate. The novel template, for example, changes the Draft name to

Manuscript. Note that when the Draft has been renamed, all interface elements that refer to it will have their labels updated.

Some templates will have "document templates" inside them, already pre-configured. This will provide you with starter documents for creating new elements within the template. You can modify the items in this folder to customise them or add your own.

7.10.2 Creating Your Own Templates

Creating custom templates is as easy as creating a new project, and if you often set up projects with the same starter items—like character sheets, keywords, or custom labels—personalised templates will save you a lot of time. Here is a list (by no means complete) of things that are commonly changed or added to custom templates:

- Custom labels & status: add the types of labels, preferred colours, and status stamps that you find useful for your projects.
- Project references: try making a list of writing resources, critique groups, and research portals that you often use.
- Character sheets or research note starters use Document Templates (section 8.5) to supply yourself with your favourite starter documents.
- Compile settings: since compile settings are stored in the project settings, you can configure these so that your future projects will be ready for one-click export (or close to it!).
- Starter Collections: standard collections as well as saved search collections can be placed in a template as well, though initially they will probably be empty.
- Starter story structure: set up your favourite story-building techniques and outlines in the Draft.

The important concept to keep in mind is that whatever you can save within a project, you can save as part of a template[3]. There are a few differences (mostly pertaining to how a template is loaded and the description and thumbnail that can be saved with it), but for the most part you should consider a template no different from an ordinary project.

Try not to worry about getting everything perfect the first time through. On average, it will take at least half a dozen new projects before things start to settle down. It is easy

[3]Technically, there is a 50 MB size limit to what can be saved as a template, but given that a template is intended as a *starter* project, you are unlikely to ever encounter this restriction.

to update existing templates with revised material; this will be covered in greater detail in the following pages.

Once you have set everything up, use the **File ► Save As Template...** menu item to start the template creation process.

In the "Template Information" box, provide the following details:

Title The visible title of the template (what will appear beneath the thumbnail in the template browser).

Category Which section to add the template to in the browser.

Description A brief description of what the template provides. You may not find this necessary for yourself, but if you intend to share the template with others, it is a good idea to explain what the template is meant to provide.

The "Icon" box on the right gives you options for setting the appearance of the template thumbnail. You can choose from a number of available presets, or if you wish to make your thumbnail stand out from the built-in templates, you can click the **Save Icon as File...** button, which will generate an image file for you based on the currently selected thumbnail. After you have edited this image in your favourite image editor and saved it as a graphic file, you can use the drop-down menu to select **Custom...** and choose the saved file from your computer.

After clicking **OK**, the template will be saved into the system, and you can delete the original project or continue working in it without affecting the template.

Retina Friendly Icons

Creating graphics that look good on both high-resolution displays and standard displays requires the use of special files, formatted with two individual images inside of it. Unfortunately the technique that Apple uses to work these special images is not a standard technology, so no normal graphics editing programs will be able to view both pieces of the file or create one. Opening and saving the TIFF file that Scrivener creates will discard the normal quality image, meaning that the resulting graphic will be too large for standard displays and be reduced by the software, causing a blurry image to appear in the template chooser.

Literature & Latte has created a free tool called MultiTIFF, available from our web site in the free downloads section[4] which can help you extract multi-TIFF files or create them. All of Scrivener's default template icons are embedded with high-resolution

[4] http://www.literatureandlatte.com/freestuff.php

and low-resolution versions of the icon. If you open them using this tool, use the File ► Export ► Both Images menu command to create two PNG files. These can then be modified in the image manipulation tool of your choice, saved, and then dropped back into the MultiTIFF window in the respective slots. Saving the file will create the dual-image TIFF file which can then be used in the project template creation tool as describe previously.

> **Custom Document Icons**: For the same reasons, custom document icons you create for the binder (section 8.6) would need to have two versions of the file if you intend to view the project on more than one computer and care about the quality of the graphics. For best quality, icons should be 32 x 32 pixels for high-resolution, and 16 x 16 pixels for standard resolution.

Template Placeholder Tags

In addition to the standard project elements you can include in a template, there are a few placeholders that you can type into the editor which are especially useful if you intend to share the template with others. For example, if making a template for yourself, you could just write in your name and address on your manuscript submission cover-page. This wouldn't work so well if you wish to share the template, though, and in fact Scrivener will check your templates for stuff that looks like your personal information and warn you if you wish to proceed. Template variables let you place a special code where text should occur, and when the template is used to create a new project, it will consult your Contacts.app for the relevant information. Many of the built-in templates that ship with the software have been set up to do this already, which is how they place your personal information into new projects. You'll find a list of available placeholders here (Table 7.1).

If you wish to edit a template which uses these variables (such as many of the built-in templates), follow these instructions:

1. Create a new project (⇧⌘–N).
2. Select the template you wish to edit.
3. Hold down the Option key, and click on the Choose... button.
4. You can let go of the Option key and give the project a name.

Holding the Option key will inhibit the substitution of these placeholders for their Contacts.app equivalents, so you can edit the template file and then save it as an update.

Table 7.1: Project Template Placeholders

Code	Description
<$template_firstName>	First name from Contacts.app
<$template_lastName>	Last name
<$template_fullName>	Combines the first and last name for you, ordering them according to system language preferences.
<$template_initial>	First letter of first name from Contacts.app
<$template_street>	Street address
<$template_city>	City
<$template_ZIP>	Postal code
<$template_state>	State or county
<$template_country>	Country
<$template_phoneNumber>	First phone number listed in your address card.
<$template_email>	First e-mail address listed
<$template_projectName>	The file name of the project from when it was created. Note this will not include the ".scriv" portion on the end.

Updating Templates

To update a template which has already been saved, follow the above instructions to create a new project using the template you wish to alter, without substituting the placeholders, make any changes you wish, and then choose **File ▶ Save As Template...**. All of the information from the original template will be filled in for you, so you do not have to worry about replicating this information every single time. If you do not need to make any changes here, just click the **OK** button and then **Yes** to confirm that you wish to update the existing template.

Note that after you have created a project from a template, the two have no further connection. Thus, existing projects that you created from a template will not be modified to reflect changes made to the template. Likewise, if you have created a project purely to update a template, there is no need to keep the project around after that point, as allowing the project to auto-save will not automatically update the template; you must choose **Save As Template...** again.

Built-in templates cannot be overwritten. However, you can create projects from them for editing and then save your modifications as a new template by giving it a new name.

7.10.3 Managing Templates

In the Template browser window, the **Options** menu in the bottom left provides the following for managing templates:

Set Selected Template as Default Changes the default template selection to whichever template you currently have selected. A yellow border will appear around the selected template. Once this is set, the next time you call up this window it will highlight that template for you.

Hide Getting Started Disables the special starter category at the top of the category list, which contains useful links to the Interactive Tutorial and the User Manual. These can also be access from the Help menu at any time, so it is safe to remove the Getting Started category once you've familiarised yourself with the program. You can also choose to reveal it again in this same option drop-down.

Import Templates... If you have downloaded templates from the Web or copied your custom templates from another computer, use this feature to import the files into the template system.

Export Selected Template... Useful for sharing your templates on the Web with other authors or for transferring templates to another working computer.

Delete Selected Template When a custom template is selected, you can use this menu item to remove it from your system. Note that the built-in templates cannot be removed.

7.11 Backing Up Your Work

Regularly backing up your work is an important part of the writing process in that it keeps your efforts safe, and while there are many external strategies for keeping your work safe from catastrophes and mistakes, the most important part is remembering to do it. Fortunately it is possible to set up Scrivener to handle most of the latter part for you.

7.11.1 Configuring Automated Backups

By default, Scrivener will automatically back up every project that you work on, whenever you close it. These backups will be stored in your user folder:

```
~/Library/Application Support/Scrivener/Backups
```

For those using the Mac App Store version, this will be in your Scrivener sandbox container instead:

```
~/Library/Containers/com.literatureandlatte.Scrivener/Data/
    Library/Application Support/Scrivener/Backups
```

[10.8 **(Mountain Lion) Only**] It is important to note that if you have changed this location in the past, when you first run the new Mac App Store sandboxed version, the target backup folder will be changed to the above. If you wish to go on using the original location you set, you will need to manually choose it again. This action gives Scrivener permission to continue doing so in the future, it cannot imply permission based on your prior settings, so this explicit step must be taken by you.

This location can be changed in preferences, and could even be set to save into a folder that is synchronised over the Internet, such as with Dropbox. Also by default, projects will be zip archived to save space and protect the internal files from external utilities or transmission errors, and Scrivener will rotate the files (delete old backups) to keep the maximum for each project under 5 backups. In most cases, the more automatic functions you activate here, the slower things will get. Finding a balance between frequent backups and usable settings will be up to you.

It is important to note that the backup system, as it ships by default, is set up with a common method of working in mind, whereby you close your project on a regular basis (say at the end of every day). If you find your own work habits deviate from that, you would do well to go over the Backup preference tab, and change the settings to better suit your style of working. Those who leave their projects open for weeks or even months at a time, will most certainly want to change the backup settings to be triggered by manual saves, and remember to run a manual save at least once a day for maximum protection.

On the other hand, if you open and close projects multiple times a day, you might find the default limit of only five copies too limiting. Mistakes you'd like to revert may have rolled off the backup list because five backups were created in the past day or two. Increasing the number of stored backups to a higher value means more drive space will be used for each project, but will ensure you have something from more than a few sessions ago to restore from.

In cases where security is a concern (if you are working with confidential files in a protected area for example) the automated backup system might present a security risk if it produces files in an unencrypted area of your hard drive. If you are working in an encrypted environment, make sure the backup location is set to also output to that area, or use the **File ► Back Up ► Exclude From Automatic Backups** menu command to restrict the sensitive file from getting backed up with the rest. Read Using Scrivener in a Secure Environment (section D.2) for more tips on locking down your projects.

Read about the various options available in the section on the Backup tab (section B.11) in Scrivener's preferences.

7.11.2 Managing Backups for Large Projects

Very large projects can conflict with what would ordinarily be good settings for the automated backup system. If a project has reached a point where backing it up automatically has become a nuisance, rather than decreasing the amount of backup security globally, consider excluding the large project from the automated system.

While the project is open, use the **File ► Back Up ► Exclude From Automatic Backups** menu command. Once this option is checked off, the automatic backup system will ignore this project entirely. It will then be up to you to keep manual backups of the project. The **File ► Back Up ► Back Up Now** feature is useful in this regard.

7.11.3 Manually Backing Up

Backups can be created whenever and wherever you want. Use **File ► Back Up ► Back Up To...** and select a backup location and filename. The "Backup as ZIP file" option in the subsequent dialogue will compress the backup project into a zip file after saving it, and is thus useful when backing up to remote storage locations, such as Dropbox. By default, backups will be timestamped, making it easy to find a precise version later on and reducing confusion over which version of the project is the most up to date (the one without a date is your primary project).

It is also possible to manually trigger the automatic backup system, even for projects which have been excluded, by using **File ► Back Up ► Back Up Now**. This will follow any relevant options that have been set in preferences, such as how many to keep, where to store them, and how many to keep around. Do be aware that since automatic backups rotate the oldest versions out, if you use this command frequently you might end up losing a backup you later want. If unsure, use the manual command to create a new backup file at a location of your choosing.

Frequent use of this feature will help safeguard your work in progress, and it is recommended that you start forming a habit of making backups whenever a decent amount of work has been committed to the project.

7.11.4 Restoring from Backups

In the unfortunate event that a backup is needed to fix a problem with your current working project, restoring couldn't be easier. As noted previously, backups are by default just Zip archived projects, a compression format that is natively supported by all modern operating systems. They are, compression aside, no different from the project you work with on a daily basis, though they will be successively older revisions, and therein lies their power. The only other thing that makes them special is that they are tucked away in a safe out of the way place, to help avoid being inadvertently modified. If you are unsure of where your backups are located, you can always open the folder they are stored

in by opening the Backup tab in Scrivener ► Preferences... and clicking the button at the bottom, Open backup folder..., to reveal them.

> **What are Zip files and how do I use them?**: Zip files are a "container" file format. You can think of container files as being a bit like boxes. You can stuff other things into a box, tape it up and store it in the attic. It takes up less space and protects the things inside the box to a degree. It makes them easier to move around because it makes lots of little things seem like *one thing*. However a box is not the thing(s) you put into it. This might seem like common sense, but a lot of people wonder why they cannot open their .zipped project backups in Scrivener. The reason is simple: so long as it is in a .zip file, it is not a Scrivener project. You would not try an eat a candy bar that had been placed inside of a cardboard box. To eat the candy bar you need to remove it from the box (and probably from its own wrappings) first. The same goes for digital files that you store inside of a digital box. They must be removed from that box before they can be used.

Once there, you'll find a number of files named respectively with the projects they relate to. After the project name there will be a sequence number or a date stamp. It is best practice to select the backups you wish to check and copy them to a temporary location, first. The backup folder rotates content, so copying a set out will ensure *these* backups do not get deleted by Scrivener's cleaning routine while you experiment. To do so, simply click and drag the selected .zip files while holding down the Option key, and drop them some place convenient, like your Desktop.

Locate the most recent backup[5] and unzip it by double-clicking on the '.zip' file. Zipped projects cannot be opened directly in Scrivener. The unzipped project is a full project and can be opened in Scrivener in the usual way. Examine it, and if it will suffice, **close it**, then remove the broken version of the project and replace it with this extracted copy. You can now clean up the temporary working area if you wish and continue working as normal. If it contains the error as well, then proceed to the next older backup, continuing until you find a valid copy.

If you find you need to only restore *pieces* of a project, you can open your current working version at the same time as the backup, and drag those pieces from the restored backup project into the live working project, from binder to binder. This can be useful if you've done a lot of work since the last backup, and have only just noticed that one file

[5]The sequence number cannot be used to determine this as Scrivener rotates through numbers. If you have not enabled the date stamp option, you will need to use your operating system to determine the modification date of the file.

had been accidentally messed up some time ago, and there are no snapshots to recover from.

Once you've finished, the files you've worked with on the Desktop can be removed at your convenience.

7.11.5 Tips for Using Time Machine

Mac OS X Leopard and greater provides the Time Machine feature, which will automatically back up your computer once every hour, and store backups as far back in time as possible, reducing the frequency of these backups the further back in time you go.

This presents a unique problem with Scrivener in that the hourly backup routine is likely to run while you are working in Scrivener. This means that Time Machine will be capturing your project while it is open and in progress, a state that could produce incomplete backups if you try to restore from that particular point in time. Frequent users of Scrivener may very well keep their projects open for weeks at a time, meaning good backups of their project will be few and far between.

There are fortunately a few tips you can use to help Time Machine work effectively with Scrivener:

1. Time Machine can be set to run manually at any time of your choosing, using the menu status icon in the upper-right hand portion of your display. You can thus control when Time Machine makes a backup of your projects, making sure they are closed first.

2. As Time Machine starts to erase hourly backups that are old, it saves only the last backups made in a single day. In conjunction with the first tip, you can make certain that your "safe" backups are retained once Time Machine starts erasing old backups, by always running Time Machine manually at the end of every day.

3. Always use Apple's Time Machine interface to restore projects. The way they are stored on the backup disk means they are in actuality held at scattered locations throughout the external backup disk. It only saves the *parts* of a project that have changed to save space. Copying a .scriv file out of the backup disk will hardly ever result in a valid project file. Apple's Time Machine tool will always re-assemble a complete version of your project by building it together from hundreds of individual backups.

> **Going the extra mile**: Time Machine is a wonderful tool for what it does, but it shouldn't be used as your sole backup for two important reasons. First, being attached to your computer at all times, it is thus susceptible to the same risks of damage and loss

> due to theft or catastrophe. Second, no backup system should be considered infallible, and thus you should have more than one method. Time Machine isn't perfect; don't let it be your single safety net.

7.12 Sandboxing and Authorised Folders

[**App Store Version**] Sandboxing is a technology introduced with Mac OS X 10.7 (Lion) intended to provide extra security. In order to enforce these enhanced security measures, Apple has ruled that all applications sold via the Mac App Store must be sandboxed. Sandboxing provides this extra security by limiting the tasks that applications are allowed to perform. One such limitation is that applications can only access files and directories that the user has granted access using an Open or Save panel.

This means that Scrivener is unable to open or access arbitrary files on your system without you telling it to. For the most part, this is fine—when you open a Scrivener project, most of the files it needs are contained within that project file and, by opening the project, you have given Scrivener permission to operate on those files, meaning that you can edit, save and work as usual. However, there are certain types of files that, under sandboxing, Scrivener will not be able to access so easily:

- *References*: Scrivener's inspector can store a list of references to external files. Because these files are stored outside the project, Scrivener may not be able to open them by default.
- *Alias files*: it is possible to import research files as aliases using File ▸ Import ▸ Research Files as Aliases... under sandboxing, Scrivener may not be able to open such files, and when you try to open such files in the editor you will see a message telling you that the file cannot be opened.
- *Linked images in text*: when inserting an image into text, rather than inserting a copy of the image, it is possible to insert it as a link to the original file on disk using Edit ▸ Insert ▸ Image Linked to File... If the original images are stored in locations to which sandboxing does not grant Scrivener access by default, these images may not appear correctly in the text, a placeholder image being displayed instead.

7.12.1 How to Grant Scrivener Access to More Files

Fortunately, there's an easy way to grant Scrivener permission to various locations on your computer so that Scrivener can still use all of these files. Here's how:

1. Use the Scrivener ▸ Authorize Folder Access... menu command to open the directory access panel.

2. To grant Scrivener permission to access external files, simply click on the **Add Folders...** button. This will bring up an Open panel from which you can select the directories you wish Scrivener to have access to. Alternatively, you can just drag files into the "Accessible Folders" list from the Finder.

3. To stop Scrivener accessing files, select the folders in the list that you no longer wish Scrivener to have access to and click on **Remove Access**.

7.12.2 Tips

When you grant Scrivener permission to access a directory, it will be able to access all of the files inside subdirectories of that folder too. Thus, you need only select the highest level folders that you feel comfortable giving to Scrivener.

The main thing to consider when adding folders to the list is which files you need Scrivener to access. For instance:

- If you tend to import many different files into Scrivener as aliases (using **File ► Import ► Research Files as Aliases...**), or if you make heavy use of Scrivener's "References" inspector list to refer to external files, then you may wish to consider adding your entire home folder to the list of accessible folders. This will give Scrivener access to all folders inside your home folder without you having to add them all separately, such as your Pictures, Movies and Documents folders. The downside of this approach is that it somewhat defeats the point of sandboxing, which is to ensure that, should any malicious software take control of your other programs, it is limited in what it can access. Thus, although adding your entire home folder is the easiest approach, it is also the least safe in the event that your computer is exposed to malware.

- If you are very careful about where you store all the files on your hard drive, and you only tend to add images to Scrivener as linked files and import a few sound and movie files as aliases, you might only need to add the ~/Pictures, ~/Movies and ~/Music folders to the list.

- Perhaps the safest and easiest approach would simply be to have a folder somewhere on your hard drive for storing your research documents, and only grant Scrivener access to that.

Essentially, though, for any files you want to be able to view in Scrivener that are not stored inside the Scrivener project, you must grant access to one of the folders containing that file. For instance, suppose you have added the file "~/Documents/Ideas/Writing/MyGreatIdea.pdf" as a reference in one of your projects.

For Scrivener to be able to open that file, you must grant access to one of the following folders:

- "~/Documents/Ideas/Writing": in which case Scrivener will be able to access all files inside the "Writing" folder.
- "~/Documents/Ideas": in which case Scrivener will be able to access all files in the "Ideas" folder, as well as any folders contained in the "Ideas" folder and any files or folders inside those.
- "~/Documents": in which case Scrivener will be able to access all files, folders, subfolders and sub-files within the "Documents" folder.
- "~/": in which case Scrivener will have full access to your home folder. (Adding this is the easiest option if you do not want to have to worry about authorising the right folders every time you add a new file, but it is also the least secure.)

Chapter 8

Setting Up the Binder

This chapter will go over the various aspects of the binder, and how to best use Scrivener's features to structure, navigate around in, edit, and write your draft. Scrivener does some things which are unusual, and generally restricted to high-powered expensive applications. As a consequence, this can result in some interesting surprises, some of which are revealed in the built-in templates distributed with Scrivener.

8.1 How the Binder is Organised

The binder is the document browser on the left of the main window where you can organise your files (by default it is coloured a pale blue when the project window is active). It allows you to structure and rearrange your work with the utmost flexibility, much like a ring-binder in the real world (hence the name). To provide one example, you can drag files from the Finder straight into the binder to import them, which you can read more about in Gathering Material (chapter 11), or from one project binder to another.

The binder is an optional element, though in most cases you'll want to rely on its organisational features at all times. To hide the binder, click the Binder icon in the toolbar, select **View ▶ Layout ▶ Hide Binder**, or press ⌥⌘–**B**. All of these commands can be used to restore its visibility if it has been hidden.

The binder is labelled at the top. Some functions will automatically replace the Binder view with a specialised listing. Project Search Results are the most common of these. This label will change to reflect whatever sidebar view is currently in use, and additionally the colour of the sidebar background will change to help notify you that you are no longer looking at the main project binder. You can always get back to the binder by clicking the small **X** button in the footer bar of the binder, or by clicking on its tab, if Collection tabs are set to visible.

8.1.1 The Three Root Folders

The binder has three default root folders which cannot be deleted or moved from the top level (although they can be renamed, moved around amongst each other, but not

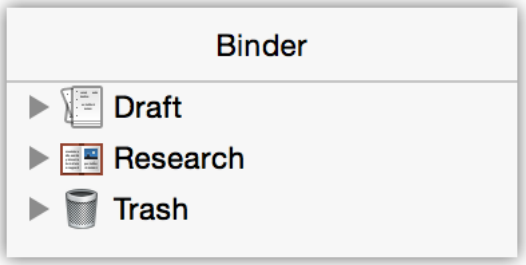

Figure 8.1: The three default top-level folders

within each other or other folders). To use Scrivener effectively, it is very important to understand the significance of these folders.

The Draft Folder

As the name suggests, the Draft folder is where you place all of the files you want to include in the actual work that others will read. How you structure it is entirely up to you—you may have parts and chapters, or you may have separate files for each scene within each chapter, or even separate files for each individual paragraph if you so wish. This user manual, for example, has each section of the manual in its own document. From this header "The Draft Folder" on to the next, "The Research Folder", is the contents of a single document nested within the "How the Binder is Organised" section, which is turn nested within the "Setting Up the Binder" chapter group, and so on.

> **My Project Doesn't Have a Draft Folder**: Some of the templates that we provide with the software have had the Draft folder renamed to something more meaningful to that template's purpose. For example the Novel template has the Draft folder renamed to "Manuscript". The Draft folder can be renamed to whatever you prefer, and in fact some people use that to type in the name of their work. The key thing to look for is the special icon[1]. Only one item can have that icon, so no matter what it is called, that is your Draft folder.

Everything that goes inside the Draft folder will be compiled into one long text file when you use the "Compile…" command from the File menu (excepting documents that have their "Include in Compile" setting unchecked). Thus, the Draft folder is central to Scrivener: ultimately, you are aiming to complete and arrange everything inside the Draft folder so that each of the elements it contains form an organic whole that can then be output as a single file to the format you require (which could be anything from a web page to a Word document to an e-book).

Because the Draft folder is what is used to create the final manuscript, it is unique in that it can only contain text and folder files. While media files cannot be placed directly into the Draft folder, images can be placed into the text files themselves, in their appropriate locations within the text, just like in a normal word processor.

A good way to think of the draft is like a long spool of paper or a scroll. In a normal word processor, this entire spool would be presented to you in a single window, with one scrollbar. Moving text around within that spool means cutting and pasting it from one spot to the next. In Scrivener, you can take that long spool of text and chop it up. These chopping points are entirely up to you—what you find useful as a writing tool. The compile feature offers a way to tape the whole thing back together into a single file, which is what you'll need to work with the rest of the world. Meanwhile the "Scrivenings" editing mode does the same thing, but in a temporary fashion, and as such is an authorial tool which makes working in smaller pieces easier, since you can at any time view and edit a larger segment of your draft as though it were a single file. These features will be discussed in depth in due course, but for now it is good know that there are no penalties for slicing and dicing up your manuscript into an easy to visualise outline.

The Research Folder

The Research folder is the default location for non-text documents such as images, PDF files and so on (although it can hold text files too). You can create as many subfolders as you wish to organise your research, or you can rename it and create other folders in the root level to hold different types of research. You aren't limited to using this one location to store material. Anywhere outside of the Draft folder a valid place to organise old drafts, notes and inspirational materials.

The Trash Folder

Whenever you delete a file in Scrivener (by pressing ⌘–**Delete**, for instance, or by selecting the **Documents ► Move to Trash** menu item), the file is not actually removed from the project but is instead moved to the Trash folder. You can tell at a glance if the Trash folder has anything in it, as its icon will appear full. To permanently remove files from the project, select "Empty Trash" from the Project menu after having moved files to the Trash folder. Note that although you can rearrange items inside the Trash folder or view it like any other binder item (and of course, you can move files into the Trash folder by dragging); you cannot create new files inside it. It is not meant to be a working area. If you wish to have a secondary holding area for files you are not sure if you want to really delete, you can create your own folder in the binder to hold them.

Files that have been placed in the Trash will have their icons ghosted. This is useful in differentiating old files, since project search results will include trashed items as well

(trashed items can also be hidden entirely using the magnifying glass option menu).

8.1.2 Scrivener as an Outliner

If, like many authors, you are used to using a word processor for all of your writings, you probably have a few habits that you'll inevitably end up bringing with you to Scrivener. One of these is working with large sections of information. In a word processor, it can be clumsy (and even dangerous) to work in extremely long files the length of an entire book, but at the same time it can also be clumsy working with sections that are too short; it means having so many files open at once and no good way of keeping all the windows organised.

A common tendency is to treat the binder like a flat list of those semi-large files, say the length of a chapter, because even just that small incremental step can be so much better than working in a word processor. But if you organise your project in this fashion, you could be missing out on a part of the Scrivener experience.

As it uses the convention of folders and files, you might instinctively approach the binder as a sort of "super-Finder", but in actuality, it's far better to think of the binder in terms of being a bonafide outliner. You might have been drawn to Scrivener for this very reason, but because of the daunting number of things that must be learned all at once, have found yourself falling back to using a simplified file-is-chapter approach.

If you have never used or even heard of the type of software referred to as an "outliner", some of this may not make immediate sense. Outliners are a fairly old software invention (nearly as old as the spreadsheet), that never really caught on as dedicated software. However the *concept* of outlining has become widespread in many programs, even including the way our operating systems show the contents of our disks. Outlining is in part a visual metaphor that declares some things as being "children" to other things, based upon their relative level of indent. If something is indented "beneath" an item, then it is considered a child of it, and that item is a parent to it. Items at the same level of indentation are referred to as "siblings". Another crucial feature of the outlining principle is that you can hide the child portions of the outline by folding, or collapsing their respective "parents", keeping your screen focussed on what you are currently working on.

You might immediately draw a connection between this and the outlines we had to compose prior to writing essays in school. Unlike those pen and paper outlines, a digital outline has the benefit of being able to contain the *content* of the work within the outline. You can imagine it as being as though each line on the paper had a stack of pages beneath it, where the portion of that essay is represented by the topic line it relates to. Thus, moving topic lines around in a digital outline moves the content *beneath* the outline for you. It is like having your very own table of contents that instead of reflecting upon your

choices, allows you to use it like a tool and make those choices with a single and easy drag and drop.

If all of this still does not make sense, I recommend going over the Quick Tour (chapter 6) provided in this manual. It approaches the concept of outlining and how the text you write can be organised into a topical tree. You might think this is all more complexity than you need, but in my experience, once the lightbulb switches on to this idea, it becomes second nature in the same way we now think of individual files on our computer as holding discrete chunks of information. You will likely find your work becomes more fluid and visible to you as you can now survey not only the major peaks of your work (chapters), but smaller components within those chapters, too.

The next section will demonstrate how Scrivener approaches this concept, and how it can allow your book to emerge as a genuine outline. That is what the binder can do for you, and unlike other word processor applications or plug-ins that tout outlining capabilities, you can use your own internal "authorial outline" which may or may not have anything to do with the visible structure (e.g. chapter names, space breaks between scenes, subsection numbering) of the book itself.

8.1.3 Folders are Files are Folders

As you no doubt instinctively know, a folder is a place where you organise files. You can create folders on your disk and then put files and even other folders inside of them to keep everything nice and tidy.

Let's throw away all of those notions for a moment and take a new, fresh look at folders, because in Scrivener, folders are an entirely different animal. Select a folder in one of your projects by clicking on it in the binder. You'll probably get a corkboard view of its child items or perhaps an Outliner view if that is how you have things set up. This is all pretty straight-forward, but what happens if you close the corkboard view?

With the folder still selected, click the currently active Group Mode button (Corkboard is the default) in the main toolbar (or use **View ▸ Corkboard**). You can tell which mode is active by its shaded background. After clicking the shaded button, you should get a blank view that looks suspiciously like an empty text document. In fact, that's precisely what it is. Type in some text and see what happens. Note that in the binder, the Folder icon has changed, and now sports what looks like a little page of paper in its corner. This means the folder itself has text associated with it.

Let's try something a little more radical. Right-click on the Folder in the binder and select "Convert to File". It should change to a stack of papers with words on them. Delete the line you wrote in that item and it will turn into a stack of paper with an empty page in front. Go ahead and right-click and select "Convert to Folder". Nothing fundamental changed here. You didn't lose any information, the only thing that changed

is the underlying type of the item, which is really primarily a visual change. When an item has other items beneath it, but it is a file, it is often referred to as a document stack, or a file group, but fundamentally it can act just like a folder does. This documentation will often refer to either folders or file groups as "containers", when the distinction does not matter.

Let's test that theory. Select one of your files in the binder. You should see its text appear in the editor as you always do. But let's do the same thing we did earlier with the Folder, click on the corkboard icon. It will be empty, but you are now looking at the corkboard for that file. With the focus in the Corkboard, press ⌘–N to make a new document. An index card should appear, just as if you were adding items to a folder. In the binder, you'll notice that the icon for this file has changed to the aforementioned stack of paper.

Perhaps you can better see why it is a good idea to think of the binder as an outliner, instead of a glorified Finder. What is all of this flexibility good for? You don't have to be thinking in terms of constrictive structural elements as you flesh out the skeleton for your book. You can just add items as you go, building out new corkboards, and changing things to folders or back to files once a structure begins to emerge.

Scrivener lets you work the way you find most comfortable. There is no need to build out a strict outline if you've always worked text-first, but if you like to work in outlines, or use the snowflake model of expansion, you can accomplish these with ease. An outline can literally emerge out of your book, the book can be built onto an outline or even a mix of the two. It's all up to you and how you work best.

In Scrivener, the concept of a Folder is fluid. It's an important concept to grasp, because organising your book will inevitably mean a hierarchy of documents, and wherever that hierarchy takes you it will take so in the form of folders, or their cousins, the document stack. The concept of hierarchy is important, because it means you can keep the parts of the book you aren't currently working on tucked away, and that means you can feel free to break things down as far as you want, no matter how many hundreds of items you make in the binder, if you sort things into folders (or file groups!) you can always keep the clutter at bay.

8.2 Binder Icons

There are many binder icons (Table 8.1) used to show the type of item, and in the case of folder and text documents, hints about its status as well—the text and folder document icons will change depending upon certain conditions. A new document that you've just created will look like an empty sheet of paper, until you add a synopsis to it, then the icon will change to an index card. If you start typing into the main text editor, the icon

will then show a sheet of paper with "words" in it. Likewise, folders will display if they have synopsis or text content (see the prior section if that comment resembled gibberish).

How you use this information is up to you. Some people like to develop their outline in stages, first building the skeleton, then fleshing out the longer descriptions in the synopses fields, and finally starting to write prose into the pieces of the outline. These indicators can help them see how far along into the process they have gone, and what areas of the draft need more work.

Table 8.1: File and Folder Binder Icons

Icon	Explanation
	Text Document Icons
	An empty text document.
	A synopsis has been added to the document, but nothing has been typed into the main text.
	This text file has text content.
	Scriptwriting documents.
	These variations of the above icons indicate that the text item contains subdocuments; child items indented beneath it in the outline.
	Folder Document Icons
	A standard folder document.
	The small index card badge means this folder has a synopsis added to it.
	A printed page badge indicates the folder has main text content.
	Any of the above icon variations can be "dog-eared". When an icon has a flap folded over in the top-right corner, that means the item has snapshots (section 15.6) saved for it.

8.3 Binder Controls

This section contains some tips for manipulating the outline in the binder.

Revealing and Hiding Elements Click on the arrow to the left of any container to reveal or hide its contents in the binder.

Use the LeftArrow and RightArrow keys to open and close containers.

In conjunction with the Option key with either of the above methods, expand or collapse an entire portion of the outline tree at once.

Reveal and Hide All "Expand All" and "Collapse All" in the View ▸ Outline ▸ submenu will disclose or reveal the entire outline at once. Expand All can be accomplished with ⌘–9; collapse All with ⌘–0. Both of these commands work in many areas of the interface where it is possible to disclose items.

Collapse All to Current Level Working in a similar fashion to Collapse All, this command will collapse the entire outline, but only up to the currently selected item's level. So for example if you if have an outline that has six levels of depth, and select a folder on level 3, running this command will completely collapse all items except those at levels one, two and three.

Hoisting Focus on one area of the binder by selecting a container and choosing Documents ▸ Hoist Binder. Documents ▸ Unhoist Binder will return the binder to its full display.

Moving items with drag and drop Items can be moved around in the binder with drag and drop. Simply pick up the item by clicking and holding the mouse button down, then drag the mouse pointer to where you wish to drop it. The binder will display a blue target underneath the pointer, showing where the drop will end up. Holding down the Option key while dragging will inhibit the "Drop on" feature, making it easier to move siblings amongst one another.

Moving items with the keyboard ^⌘–Arrow keys can be used to move an item around spatially in the binder, step by step. Up and down will increase or decrease its placement in the outline, while left and right will promote and demote the item. For example, selecting a folder that holds the contents of chapter eight, and pressing ^⌘–DownArrow will move it down one spot, swapping its position with the chapter below. It will now be chapter nine, and the other chapter will be eight.

Keyboard navigation The arrow keys by themselves will move the selection around in the binder.

- Up and down arrows will move up and down the outline.
- Left and right will ascend and descend the hierarchy. When a container is selected, the first time you press LeftArrow, a container will be closed, the second time will jump up to the next parent.
- Combining the Shift key with the UpArrow and DowArrow keys will select multiple items in the chosen direction.

Advanced keyboard navigation ^⌥UpArrow and ^⌥DownArrow will jump from one visible container to the next. If a container is hidden from view, it will not be selected. When "Treat all documents with subdocuments as folders" is enabled in the Navigation preferences, file groups will be included when navigating this way.

⌥UpArrow and ⌥DownArrow jump to the top and bottom of the binder. Note this is different from the Home and End keys, which will only scroll the view, but retain your current selection.

Copying Items While ordinary copy and paste functions are not allowed in the movement or reproduction of binder items, copying a selection of items can be useful. In most cases this will place a list of names on the clipboard. When copying images from the Binder, the graphic will be placed on the clipboard and can be pasted into rich text editing fields, such as the main editor, project notes, and document notes.

You can effectively copy any number of items by duplicating them with the Documents ▸ Duplicate ▸ sub-menu. They will be copied in place, and you can then move them to their intended location.

8.4 Using Collections

Collections are a way to augment the binder method of organisation, and they will temporarily supplant the binder view in the sidebar, allowing you to reorganise or arrange collections of items in a flat list, regardless of their position in the actual project or hierarchal relationships with one another.

You can think of collected items as being a bit like an alias, or a shortcut to that item. Changes made in the editor to items within the collection will also be made to the original item in the binder, it is better to think of them as being the same thing. A single item can be in one collection or in many. Each instance will point back to the same original item in the binder.

When setting up a new project, you might like to create a few starting Collections to help organise material as it flows into the project, or as you generate new outline ideas. If you find yourself creating a sequence of common Collections at the start of every project, you might want to design a custom template (subsection 7.10.2) for yourself. Here are some example uses for Collections:

- Experiment with an alternate scene flow without disrupting the original layout.
- Collect all scenes which still need to be edited.
- Designate items which you wish to share with another author using one of the various syncing methods.

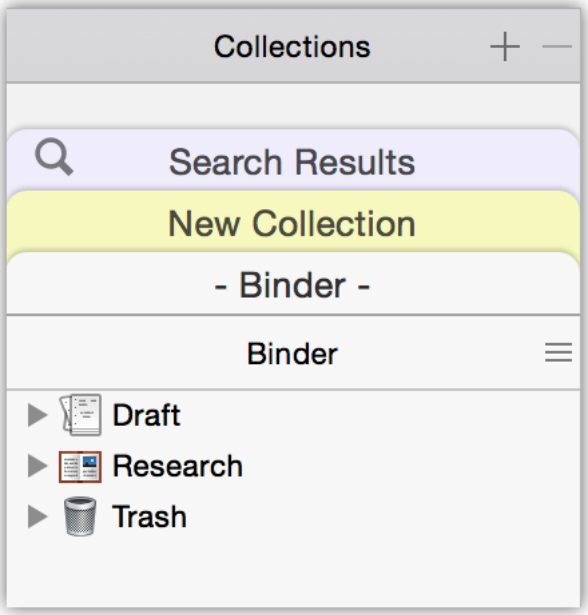

Figure 8.2: The basic Collection interface with the binder selected.

- Search for any occurrence of words you tend to overuse. Saving a search that looks for any word amongst the listed words, clicking on this tab would highlight all of the problem words, listing only those sections of the binder that contain them.
- Create a special compile group with an alternate selection and order than a typical compile of the draft.
- Store saved searches for future use, or to monitor workflows.
- Set aside groups of documents in a day-by-day schedule, removing them from each daily collection as you address them.

8.4.1 The Collection Interface

To reveal the Collection interface, click the Collections icon in the toolbar, or select **View ▶ Collections ▶ Show Collections**.

Each tab represents a single collection. In the screenshot (Figure 8.2), the "Binder" tab is selected. While not a true collection, this is how you can navigate back to the Binder after viewing a Collection. Click on any tab to select it; when a tab is selected, it will re-colour and ghost all of the tabs below it to help indicate which is the active tab. Additionally the header bar will change its title to the name of the tab. You could hide the Collections interface at this point and continue working in that tab. The background

colour of the sidebar and the header bar will help remind you that you are not in the full Binder.

When first created, Collections will be assigned with an automatically generated colour, but their tabs can be recoloured by double-clicking the colour chip to the right of the label. Likewise, double-click on the title to change the name of a collection. The Binder and Search Results tabs cannot be renamed or permanently removed.

The are three main types (excluding the Binder) of Collections:

1. Standard Collections: allows you to freely add and remove items as you work. This is the most flexible form of Collection.

2. Search Results Collection: a special collection type that cannot be removed. Any project searches will have their query and results stored in this collection. This is persistent between project sessions.

3. Saved Search Collections: indicated with the same magnifying glass used for Search Results, on the left side of the label; they will be dynamically populated with a search query every time you view the tab.

> **Disappearing Collection Tabs on Lion** On account of the new behaviour of scrollbars in Lion's default setting, it may not be obvious that the Collection tab interface can be scrolled, causing tabs to mysteriously disappear. If you do not like this behaviour, you can change your Mac's system settings in the General preference pane to always show scrollbars.

8.4.2 Setting Up Label & Status

Creating a new standard Collection is as easy as pressing the + button. Any items that you have selected in the active view (including the Binder, other views, or even other Collections), will be automatically added to it.

Once a Collection has been created, you can add or remove items to it, so it is not critical to select everything before you create it. There are several ways to add items to a collection:

- With the tab interface revealed, use drag and drop from any view including the binder. If you hold over the target tab for a moment, Scrivener will switch to the tab allowing you to drop the items precisely where you want to place them in the list.

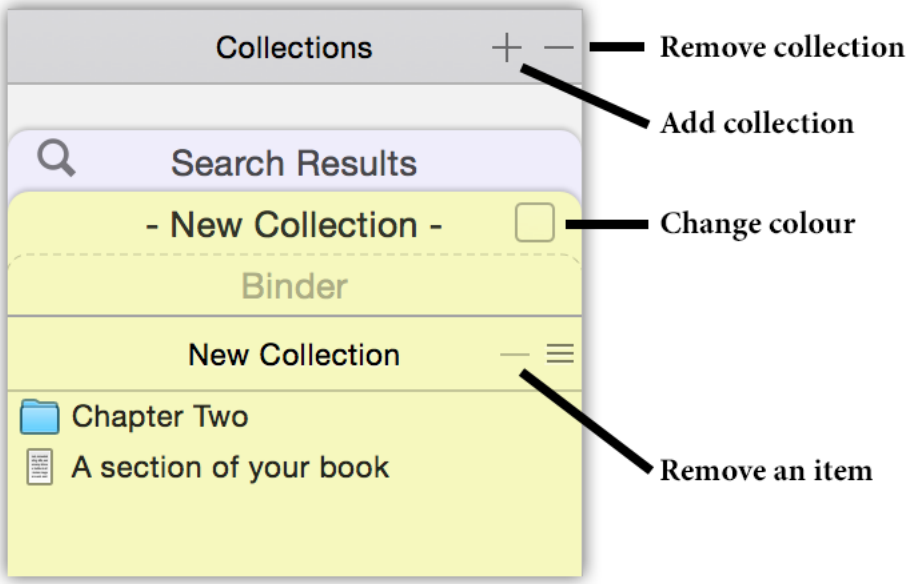

Figure 8.3: Standard Collection Controls

- Using the **Documents ► Add to Collection ►** sub-menu, any selection of items, including the active item that is being edited or viewed in the editor, can be added to the bottom of a collection list.
- In the binder sidebar, you can right-click on an item(s) to access the "Add to Collection" menu.

In all cases, if an item already exists in the collection it will not be added again, and its original position will not be changed, so it is safe to err on the side of "over-selection".

Remove items by selecting them, and then clicking the - key (Figure 8.3) in the lower title bar, or by simply pressing **Delete** on your keyboard.

Delete the entire Collection by first selecting the tab you wish to remove, and then clicking the **-** button in the upper title bar.

Items can be re-ordered within the list using click and drag, or the same **^⌘–ArrowKey** combinations used in the Binder. Since there is no hierarchy in a Collection, you will not be able to promote or demote items.

Collections take advantage of the same view modes that containers do. Click the coloured header bar title (not the tab) to view all of the contained items using the default view mode. This can be used to view Scrivenings, Corkboard (freeform and linear), or Outliner. You can also remove items from a Collection by deleting them from these views.

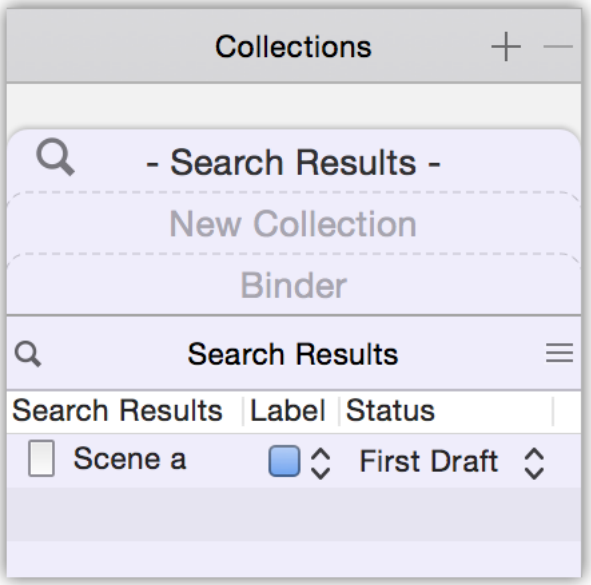

Figure 8.4: The special Search Result collection

> **Where new items go**: In Standard Collections, you may create new items using all of the ordinary tools available for doing so. Since collections are uncoupled from the Binder structure in every way, new items will be placed into folders with names corresponding to the name of the collection they were created within. These folders will be created at the top level of the binder, at the bottom of the list, above the Trash. An example might be a new text file called "Joseph" in the *Characters* collection. When you return to the binder, you will find a new text file called "Joseph" in a folder named "Characters (New)".

8.4.3 Search Result Collection

If you've used Project Search, then you've been using a Collection without perhaps realising it. Search results are placed into this special collection every time you run a search, and the criteria of your search are saved into it along with the project. This means the last Project Search that you ran will always be available to you, even after you have rebooted your computer. Like the Binder tab, the Search Result tab cannot be deleted, and there is always one included with every new project, even when no search is currently in use.

Both the built-in Search Results collection, and Saved Search Result Collections, allow you to view the label and status of the matching items, making it convenient to enact workflow changes without using the outliner. You can toggle the visibility of these columns by right-clicking in the column header area and selecting **Show (Hide) Label \&**

Status Columns.

When you click on the Search Results tab, the previous search criteria will be loaded into the search bar tool, allowing you to further tweak the results if you desire. Since search results depend upon the search criteria, you cannot manually add, remove, or change the order of items from the Collection, as you can with the standard type.

8.4.4 Saved Search Result Collections

As with the Search Results tab, saved searches have a magnifying glass in the corner to indicate that it is a special type of Collection which automatically generates its contents for you. Saved searches are created by clicking the magnifying glass in the Project Search tool, and selecting **Save Search as Collection...**. When saving from a Keyword or Label search, the associated colour of the meta-data you are looking for will be used for the tab's colour as a convenience

If the Collection interface is hidden, the sidebar background will change to a random, automatically generated colour, and the header bar title will adjust to the name you provided. To leave the saved search, you can either click the close button in the sidebar footer, or enable the Collection tab interface and select another tab.

The contents of a saved search are dynamic. Every time the tab is clicked on, the saved criteria will be checked against the current state of the project. If you wish to "freeze" a list for later reference, there are two ways you can do so:

1. Convert the saved search to a standard type: This will destroy the saved search, so only use this method if you no longer need the search criteria. To convert a saved search result to a standard collection, select the tab in the sidebar, and use the menu command, **View ► Collections ► Convert to Standard Collection**.

2. Copy the contents into a new collection: This is quite easy to do. Simply select all of the items (**⌘–A**) in the search result list, and click the **+** button in the collection header bar to create a new Standard Collection from the current selection.

Saved Search Collections can be removed in the same fashion as Standard Collections, by selecting the tab and clicking the **-** key, as shown here (Figure 8.3)

> **Limitations**: Searches performed using the "Binder Selection Only" option cannot be saved for future use, because the binder selection is a temporary state which changes whenever you click in the binder. If you find yourself unable to save a search, make sure this option is disabled.

See Also:

- Project Search (subsection 21.1.2).

8.4.5 Back to the Binder

Often you might wish to know the overall disposition of a collection in the binder, or where a specific select few items in the list are located in the overall structure of the project. The **View ▶ Reveal in Binder** menu command (⌥⌘–**R**) works from collections and search results. Since this command can be used on many items at once, it makes for a handy way to see what *isn't* in large collections.

It is possible to select some or all of the items in a collection and instruct Scrivener to gather them all together into one spot, based on the order of their appearance in the collection. There are several ways of doing so.

- Drag & Drop: you will need to have the collection tab interface visible. Simply select the files you wish to gather, drag them to the binder tab and hold for a moment. The binder will activate, and you can drop the selection wherever you please.

- Drag & Drop with the Option key: works identically as above, but instead of moving the items into the drop location, they will be duplicated.

- **Documents ▶ Move ▶ To ▶** sub-menu: select the items you wish to gather together, and either use the main application menu or the contextual menu to move them to a selected item or container. This action will operate in the background, leaving your focus in the Collection.

- Select all of the items that you wish to collect together and use the **Documents ▶ Group** command to assemble them into a new folder in the Binder.

Experienced users of outliner style programs will recognise this ability as "mark and gather". The marking phase is done by assigning documents to a collection. Moving them back out to the binder then gathers them quickly into one focussed spot. This can be an extremely useful technique for some workflows.

It can also be useful to experiment and implement an experimental text flow. If a chapter or section just doesn't seem to flow right, you can quickly create a new collection, Option-drag the container of items to the new tab, and then re-organise the flow using the collection's ability to view itself as a corkboard or outliner, then reviewing the text with scrivenings mode. Once you are satisfied with the new layout, select all of the subfiles and drag them back into the original folder in the binder using the above method. The items will be re-organised for you in the actual book structure.

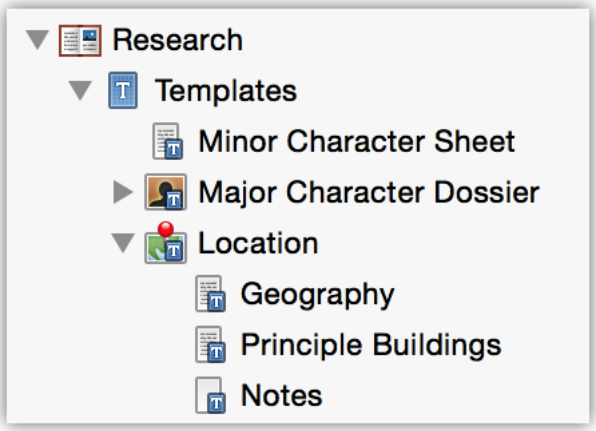

Figure 8.5: Example Document Template Structure

8.5 Document Templates

Document templates give you the ability to designate a set of items that can be used to create new items with stock boilerplate settings and text. Much as with project templates (section 7.10), everything that you add to a document template item will be added to new items created from it. This includes the various meta-data, notes, keywords, references, synopsis, and of course any text. Additionally, folder items designated as templates will create copies of their children too, allowing you to quickly generate complex repeating structures. Certain special features that can be assigned to items will be carried along as well, such as default subdocument types (subsection 8.5.5) and locked view modes (subsection 9.4.1).

Document templates do not provide forms that you can fill out with the tab key. They are simply a convenient way to reproduce ordinary folders and files, with their meta-data intact.

8.5.1 Setting a Document Template Folder

Templates are designated by placing them into a special folder, marked as being a template container (which can be a folder or file group). Each project can only have one template container at a time. To set it up, select the container of items you wish to use as templates, and use the **Project ▸ Set Selection as Templates Folder**. The binder icon will change to indicate its status, and all of the items indented beneath it will have a small blue "T" badge added to their icons.

Some of the built-in project templates will come with a template folder already set up for you. To use your own templates folder, first clear the default setting (described in the

next section). Optionally, you could just remove the contents of the default, and fill it with your own templates.

If you find yourself doing this often, you might wish to create your own project template (subsection 7.10.2) so that everything is set up the way you prefer when you start.

8.5.2 Clearing Document Template Folder

To clear the document template folder, use the **Projects ▸ Clear Templates Folder** menu item. This action will not delete anything in your project. It will merely remove the special status from the designated templates folder. Template folders can be placed (or designated) anywhere in the binder, even in the Draft (though do note that if you do that, they will likely show up in the final manuscript, and toggling them to be hidden would lead to generated items being hidden by default, as well).

8.5.3 Creating Document Templates

Creating new templates is as easy as creating a normal file or folder item in the binder. There is nothing special about these items, save for their position within the designated templates folder. Anything that you add to the templates folder will be available for usage in the project.

It is possible to pre-configure all aspects of an item. The only thing that will not be carried over is Snapshots. The latest version of the item's text will always be used and any Snapshots will be ignored.

8.5.4 Using Document Templates

Once a container is designated as a Template Folder, and items have been added to it, you can generate copies of these templates elsewhere in the project via the **Project ▸ New From Template** sub-menu, or by using the green **+** button in the toolbar. The items in this menu will be arranged precisely as they have been organised in the binder, with groups becoming further sub-menus. As with the other commands that create new items, they will be created in relation with the current binder selection.

- If you select an entire group from the menu, it and all of its child items, will be created in the selected position.

- The first template item in the template folder is a special spot as it will be given a keyboard shortcut, ⇧⌥⌘–**N**. Using this shortcut, you can easily create this special item on the fly, so if applicable, choose the top position for a template you will be using most frequently.

- As with project templates, once created, document templates have no connection with the item that they were modelled after. It is safe to delete a template that is no longer being used to create new items. However, you might wish to move the template item to another location in the Binder, outside of the template folder, in case you ever need it again.

It is also possible to assign your own keyboard shortcuts to other items, using the instructions in Customising Keyboard Shortcuts (chapter 28), but note that these shortcuts will only apply to those projects which have identically named templates.

8.5.5 Default Subdocument Types

Depending upon the type of document templates you design for your project, it might occasionally be useful to set up a folder so that when you create new items within it, they will be created as a particular template type, rather than the default text file. Quite simply, assigning a default type to a container will cause all of Scrivener's ordinary methods for adding text files to generate using the declared document template instead. This extends to all of the actions you can take to create new blank items, but no actions which would result in an item that already has content. For example, if you press ⌘–N while looking at a corkboard that has a default type assigned to it or using **Append Selection to Document ► New...**, then the new card will be created as that default subdocument type. If however you import a pre-existing word processing file into this folder, then it will not use a defined default type, but will instead just be ordinary text.

To set a default subdocument type, use the **Documents ► Default New Subdocument Type ►** sub-menu to select from the standard template selection menu, as listed in the main **Project ► New From Template ►** sub-menu.

A simple example of how this might work is if you created a "Note" document template, which sets up some boilerplate questions and answers, and has a custom icon set to use the yellow note icon. Now, if you create a folder called "Notes" in your Binder, you might want to assign the "Note" document template you've created as a default type for new items created there.

> **Watch Out for Folders**: In some cases, the default subdocument type that you've specified might in fact be a folder. Since Scrivener considers text files and folders to be synonymous, this is not too big of a deal. However it can result in a slight inconsistency with the typical behaviour for creating new folders. Ordinarily when you create a new folder with another folder selected, it will be created as a sibling to that folder. In the case where the default subdocument type is a folder, it will be created as *child* and assigned that type.

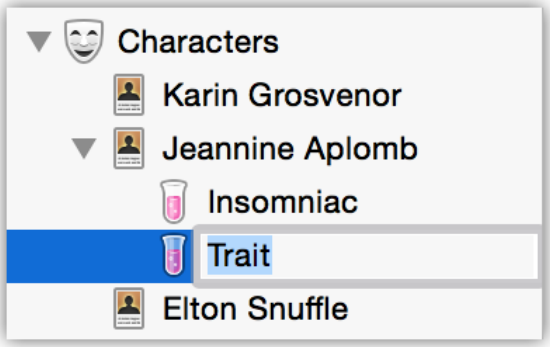

Figure 8.6: Default Subdocument Type Demonstration

Besides the already mentioned exception for imported and otherwise automatically generated documents with content (such as those imported, created via clippings or appended text), there are other cases where items are never created using the default type:

- Where pre-existing items are moved into the folder. This should be considered a feature, as it allows you to mix types later on and override the default by creating an item elsewhere and then moving it in. It also holds to the rule of thumb where items with content do not acquire a default type.
- When a different type of item is specifically called for. If you use the **Project ▸ New From Template ▸** sub-menu to specifically create another type, create a folder, or if outside of the draft import a web page, the chosen type will overrule the default type. The default type only applies where a new blank text item would be created.

Cascading Default Subdocument Types

Each folder (or container, if a file has subdocuments) can have their own default type assignment. Where sub-folders are concerned, the default type "cascades" into all sub-folders, no matter how deeply nested, they will acquire that default as well. The exception to this is when a sub-folder has its *own* default template type assigned to it.

In the provided illustration (Figure 8.6), a container called "Characters" has been set up to use a "Character Sheet" template type as its default subdocument type. Whenever the user adds a new item to this container, it automatically acquires the "Character Sheet" template style (in this case, a folder with a custom icon). If that was all that had been configured, these character sheet folders (you can tell they are folders because they have disclosure arrows even though they are empty) would create further character sheet folders beneath them, because of the cascading rule. However in this case the user has set up

the Character Sheet template to use the "Trait" document template for its subdocuments by default. Thus it *overrides* the default supplied by the "Characters" folder above it. Any new items created instead the character sheet folder will automatically be Traits.

This demonstrates a more complicated example of this feature's usage. Most often, you'll just want to create a particular type of item within a certain container and that will be it. But if you wish to set up more complex arrangements, remember that the subdocument default setting is itself something that will be duplicated along with the rest of the template type, and thus allow quick scaffolding of common types. Another practical example might be Part folders that automatically create Chapter folders with a blank scene, and those Chapter folders automatically creating new scenes within them.

8.6 Custom Icons

Custom icons can be used to embellish the appearance of your binder items. All items can have a custom icon, even the three root folders: Draft, Research, and Trash. Scrivener comes with many icons right out of the box.

There are a few ways you can change the icon for an item:

- Click on the item in the Binder, Corkboard, or Outliner, and use the **Documents ► Change Icon ►** sub-menu.
- Access the item's contextual menu, and use the "Custom Icon" sub-menu.
- Use the quick icon selection menu by holding down the Option key when right-clicking.
- Icons can be set to many items at once by selecting them all in the binder, and right-clicking on them to choose **Change Icon**, or by using the Documents menu.

You can also set the icon of the document you are currently editing in the text editor using the main Documents menu. When using Scrivenings mode, this will only impact the document that your cursor is within.

To reset an icon to its binder default (remove a custom icon from an item), select **Documents ► Change Icon ► Reset Icon to Default**, at the top of the menu.

> **Setting an icon over and over?** If you find yourself using a particular icon frequently, you might wish to make a document template (subsection 8.5.4) for it, so that you do not have to constantly re-apply icons to new files or folders.

The icon menu is organised into two major parts. The first part displays up to five of the most recent icon selections you've made, so you can quickly use them again (if you have never used the feature before, you won't see this yet). The recent icon list is retained between sessions, and is universal to all of your projects. The second section in the menu displays *sets* of icons, grouped together to save space, and finally all of the remaining ungrouped icons that are available.

8.6.1 Creating and Managing Custom Icons

It is easy to make your own icons, even if you do not know how to edit images in a graphics program. Load the icon manager with **Documents ▸ Change Icon ▸ Manage Icons...** and click the **+** button in the lower table, under "Icons in Application Support" (we'll explain what that means in the next section). Find the graphics file you wish to convert to an icon on your disk with the file chooser, and click the **Open** button. If the graphics file is larger than 16x16 pixels, Scrivener will automatically resize it for you. You can also drag and drop images into the list to import icons.

> **What to do about strange results**: After you click the **Open** button, what you see in the icon manager is what you will see in the binder and other views. If the icon looks "strange" to you in this window, then it will look strange everywhere. Common problems are files with lots of white or transparent padding around them. Open these files in a graphics editor (even Preview will do for simple cropping) and use the software to cut out the excess padding. You can remove a bad attempt and add it again using the above instructions until it looks right.
>
> Another common problem is a white background. This may or may not be obvious in the icon browser, but will look bad in the binder. Removing a white background is more difficult. If you lack the expertise to do so, try selecting another icon that is similar, from a different source. JPEG images cannot have transparent icons, so 24-bit PNG and TIFF files work best.

Icons can be dragged between the "Application Folder" section and the "Project Package" section individually, or *en masse*. Icons in the lower list will be available to all of your projects; past, present and future. Icons in the top list will only be visible to *that* project; however they can be duplicates of icons that are in the lower list. If you've created an icon for a project, and later decide you'd like to use it everywhere, you can simply drag and drop it into the Application Support table. If you try to drag an icon with an identical name from one list to another, you will be informed of the collision, and confirmation will be required before it is replaced.

When Scrivener loads your project, it checks this location (which is printed below the table in this sheet for your reference) for matching icon names, but before doing so,

it checks within the project package itself first. This means any duplicate names (not graphics) in the top list will override any names in the bottom list. This way, you can set project overrides.

> **Optimum Icon Sizes**: The "native" size of a Scrivener icon is 16x16 pixels (32x32 for Retina screens), but many of the built-in Scrivener icons are a few pixels smaller than that, to account for a subtle shadow around them. Therefore if you want your custom icons to fit in with the stock icons, it is best to size them a few pixels smaller than 16 pixels square, but leave the graphic file that large; padding the extra space with transparent pixels. For some types of icons, it will not be necessary to tweak the icon to fit in like this, so you can easily leave it larger.
>
> Also note that if you intend for your icons to look good on both Retina and standard displays, you should read Retina Friendly Icons (section 7.10.2) for tips on how to do so.

8.6.2 Working with Mac Finder Icons

Many icon packs for the Mac are distributed as Finder icons. If you would like to use an icon that you have seen in the Finder, the easiest way to create a custom icon for it is to follow these steps:

1. Select a file in the Finder that has this icon.
2. Press ⌘-I to get info on that file.
3. Click once in the small icon in the upper-left corner. You will see a halo surround it when it is properly selected.
4. Press ⌘-C to copy the icon.
5. Open Preview.app and press ⌘-N to create a new file off of the clipboard.
6. You will probably see several options to choose from in the sidebar, selecting the 16x16 version, or the next smallest available icon.
7. Press ⌘-C once again to copy the single icon choice; and then ⌘-N to create another new document from the clipboard.
8. Now you can save this icon to your Desktop or some other convenient location. It is best to use the PNG or TIFF format, leaving the "Alpha" box checked.

8.6.3 Creating Your Own Icon Sets

Icon sets are generated simply by naming conventions. How you name your custom icons will determine whether or not they are grouped together as a set in the custom icon menu. The format for this naming convention is "Category (Icon Name)". A practical example of this from one of the built-in sets would be "Flag (Blue)". This causes this particular icon to be grouped together with the Flag set.

8.6.4 Custom Icon Portability

The final thing to consider is whether or not icons will be visible off of your computer. If you've been adding icons to "Application Support" all along, if you take your project file to another computer, when you load it all of those icons will disappear. They will still be assigned, but since Scrivener cannot locate any replacement icons, they will simply be ignored until you return to the original computer.

So when working on multiple computers, it is good idea to drag your Application Support list into the Project Package list. This way your icons will be available wherever you go, and once you are on the second computer, you can drag them back into *its* Application Support list, installing them on that computer as well. Now you will no longer need to put them in the Project Package list for them to appear on both systems.

When collaborating with other individuals, it is a good idea to install custom icons into the project package. This way everyone can see and use them.

Chapter 9

Navigating and Organising

In this chapter, we will discuss the project window itself, and the many tools that are available for moving around within this window, controlling its appearance, and adjusting how it behaves when you click on things within it.

9.1 General Navigation

Moving around within a large project is an important part of writing efficiently, especially in an application like Scrivener, which lets you cut your work up into small pieces. An average book might have anywhere between several dozen and a few hundred sections arranged into many containers. Along with the binder, there are two primary tools for making sense of all of these pieces: The Outliner (section 12.2) and The Corkboard (section 12.1), covered later in this portion of the book. For now we will concentrate on how to use them as navigational tools, and as your skills increase, you'll find other uses for them beyond mere navigation.

Beyond the three primary navigation tools, there are a number of other features which can be used as needed to enhance the main three, and these will be covered in this section as well.

9.1.1 Navigation Specifics

Both corkboard and outliner views have an Auto-Load feature which when enabled, will load any selected item(s) in the other split, but only if the editor has already been split. The Auto-Load button (Figure 9.1) is located in the footer bar, and will appear blue when switched on. This action will use the current default view mode for that split.

Figure 9.1: The Auto-Load button will appear blue when activated.

Additionally, for corkboard and outliner views, you may Double-click on any icon (to the left of the title itself) to load that document in the editor using the current view mode.

⌥⌘–O to load the selected item in the editor, or when more than one item is selected, isolate the focus on just the selected items using the current view mode. This latter usage is quite useful when you have a large Corkboard or Outliner, and wish to quickly show only those items you've selected. ⇧⌘–O on the other hand will load the selected item(s) in the other editor split, opening a new one if necessary.

9.2 Go To Menu

In most cases, it will be easiest to use the binder, or the group views in the main editor, to select and navigate to various components in your project. There will be times when this is not true. If you prefer to work with the binder hidden, are working in composition mode, or just would rather not drill down to a particular spot solely to select an item, the Go To menu is an easy substitute for other methods, and in some cases can even be faster than traditional navigation. The Go To menu always takes action on the activated split if relevant, even when it has been locked. You can alter this behaviour by holding down the Option key, which will load the selected item in QuickReference panel, instead.

There are three locations where this menu appears:

- The main application menu, **View ▸ Go To ▸**. Unlike the contextual variants of this menu, it will always show the entire binder even when the active split is in a Scrivenings session. It additionally provides four shortcuts for navigating within an editing session:
  - Previous Document (⌥⌘–**UpArrow**): Selects the previous item in the binder, no matter what the relative depth.
  - Next Document (⌥⌘–**DownArrow**): Selects the next item in the binder, no matter what the relative depth.
  - Enclosing Group (^⌘–**R**): Selects the parent container which the currently selected item is within; in effect traversing up the hierarchy, much as if you had clicked on the parent container using your mouse in the binder. The current view mode preference will be used if possible. If you are editing a section of the Draft using Scrivenings mode, then using this command will expand the Scrivenings session to include the parent and its siblings, just as if you had clicked on the parent; if working on a corkboard, the next corkboard higher up will be viewed. In cases where the current view mode conflicts with the selection, corkboard will be used as a fall-back.

An example might be if you have scrivenings mode enabled as your default view mode, but you are currently viewing an image or PDF. Since scrivenings does not apply to these items, this command will switch you over to corkboard.

- Editor Selection (⌘–4): Isolates the current selection (or active document in a Scrivenings session) in the editor in single text mode, no matter what type of item it is. Useful for viewing folder text, or eliminating everything else in a Scrivenings session so you can focus on one piece.

❧ Via clicking on the item icon in the editor header bar.

❧ In composition mode, the Go To menu is accessible in the control strip along the bottom of the screen, as a button.

In all cases, both sub-folders and items can be selected. When a sub-folder is selected, it will be loaded with the default view mode (Corkboard by default), and when using the button in composition mode, it will load the text of that item. The header bar menu is provided as a convenience, and more readily applies the navigation action to a particular split, whereas one must first ensure the right split is active before using the application View menu.

9.2.1 Navigating Within Scrivenings

When the active split contains a Scrivenings session (section 15.9), the "Go To" menu will be transformed into a table of contents for that session, letting you quickly jump to sections exclusively within the session. The exception to this is the main application "Go To" menu, which will reliably show the entire binder under all circumstances. The header bar menu will always show the contents of the current session for that split.

See also, Editing Multiple Documents (section 15.9).

9.3 Hoisting the Binder

When you need to focus on one specific area of the binder, you can use **Documents ▸ Hoist Binder** to isolate the currently selected container. Only one item can be selected, and it must be a container (file or folder).

When a container has been hoisted, the appearance of the binder will change[1]. The header bar will be replaced with the name of the container that has been selected, including a small icon representation of its type, and the list of items available will be reduced to only those items which are children of the selected container.

[1] action-hoisting.png

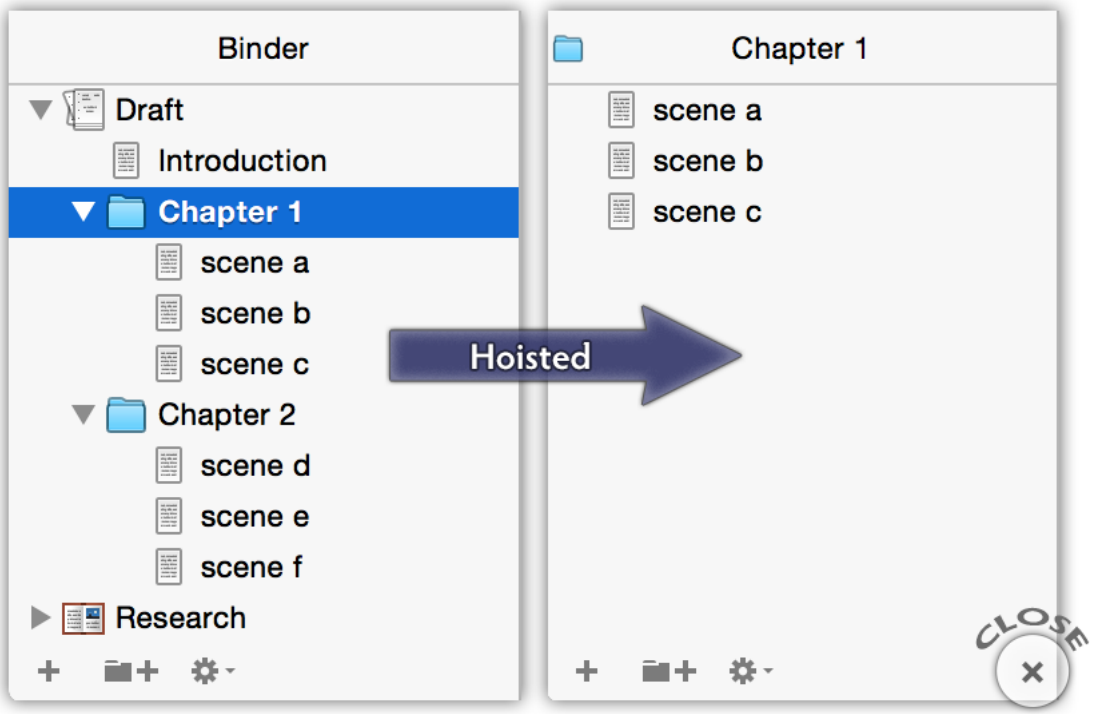

Figure 9.2: Hoisting: Before and After

If you wish to select the container itself (for example to view Scrivenings, or its corkboard), click on the header bar. Once selected you can choose the view mode you prefer, if necessary.

To return to standard Binder view, either click the indicated X button in the footer bar of the binder area, or select the menu command, Documents ▸ Unhoist Binder.

> **How to Hoist the Outliner**: the Outliner view in the editor does not have a dedicated feature for hoisting. This is because it is an automatically hoisted view by design. Simply clicking on a sub-folder in the binder would be the same as viewing its container folder contents and then hoisting that container from there. Sometimes however the binder item you wish to hoist to is not readily available—you might be working in a Collection at the time, for instance. It is possible to hoist or isolate individual items in an Outliner view. Simply select the container or items you wish to isolate, and use the Documents ▸ Open ▸ sub-menu, selecting the editor you are currently working in as the target, or press ⌥⌘–O. This will form a Multiple Selection, and once you are done with it you will be able to use the history buttons in the header bar to get back to where you were.

9.4 Controlling Binder Integration

With a few exceptions, whatever you click on in the binder sidebar will be automatically loaded into the active editor. This can become more complicated when there is more than one editor split in use. When this is the case, the split which has its header bar highlighted in blue (or graphite, when the dark grey theme option is in use) will be the one that receives binder clicks. The inverse of this is however not true. The binder will not automatically select items based upon what you are viewing in the editor. The editors can drift from the original selection quite easily. All you need to do is use the history buttons, double-click on an index card icon in a Corkboard, use the **View ► Go To ►** menu, click on a Scrivener Link in the text, and so forth. There are three features available which change how clicking in the binder sidebar work.

9.4.1 Locking the View Mode

Which view mode to use when selecting a group of items is determined by the last view mode (chapter 12) you selected. If you click on a folder and switch the view mode to Scrivenings, then you later select five separate items together, they will be shown using Scrivenings as well. In most cases this works quite well as we tend to do tasks in bunches. Sometimes however, you might wish to designate a certain container as always being shown using a particular view mode, no matter what the default may be.

To set a container (defined as a folder, file group or collection) to use a particular view mode, simply use the **View ► Editor ► Lock Group View Mode** menu command on the active editor you wish to freeze. If the toolbar is visible, you'll see a small lock icon added to the segmented control for this. So long as the view mode is locked, you can change the view mode of this container and this activity will set Scrivener to remember the last mode you used within the container.

1. A folder (A) is set to have its group view mode locked while in corkboard mode
2. Another folder (B) is clicked on, and the view mode changed to outliner
3. When switching back to folder A, the view mode returns to corkboard
4. Viewing folder B will return to outliner mode
5. Now if folder A is viewed again and changed to Scrivenings, folder B (and the rest of the project) will use Scrivenings mode as per normal. Additionally, this action will *also* lock the selected view mode to folder A. It will now use Scrivenings mode by preference instead of the corkboard.

There is no limit to the number of containers that can be locked to a view mode in a project. The setting is unique to that container and does not impact any other containers.

> **Locking the View is an Integral Setting**: This means that if it is duplicated that setting will be carried along with the new duplicate. If this setting is applied to Document Templates (section 8.5), then each new instance created using that template will also bear that setting. However from that point on these new containers will maintain their own settings; changing one will not impact the others that came from it, or vice versa.

9.4.2 Locking the Editor

Any editor split can be locked with **View ► Editor ► Lock in Place**, or ⌥⌘–L. The header bar for the editor will be changed to a dusty red colour which persists even when the split is not active. Either or both splits can be locked independently. This feature can also be accessed from the header bar icon menu (subsection 14.1.2).

Locking the editor prohibits *external clicks* from impacting the locked editor. This means that clicks in the binder, corkboard or outliner (when Auto-load is enabled), will be blocked. This makes it easy to keep your editing session while you use the binder or these other views to explore the outline. What locking will **not** do is prohibit actions taken from within the editor itself. This includes use of the binder navigation arrows, history buttons, manually dragging items to the header bar, or menu navigation commands. Locking is meant primarily to keep the interface from taking actions that it would ordinarily take automatically. Locking will not inhibit intentional actions that *you* make. These actions (as with the **View ► Go To ►** sub-menu) will navigate as requested and cancel the lock state.

A common exception to this rule of thumb is when it makes sense to scroll your view, or select items within it, based on an external selection. If the editor is a locked Scrivenings session (where more than one text file is viewed at once), then if you click on a portion of the outline that is contained within that session, Scrivener will scroll to the right spot for you. In a large locked corkboard, you can jump to individual cards by clicking on them in the binder. Again, this alternate method of navigation only works when the item you click on in the binder sidebar is already contained within the locked view. You will need to unlock the editor in order to use the binder to select items outside of the current view.

An alternate behaviour can be set in the Navigations preference tab (subsection B.6.3), "Binder selection affects other editor when focussed editor is locked". Ordinarily if the editor split you have active is locked, nothing will happen when you click on things in the binder. With this option enabled, the binder click will be loaded in the opposing editor split if one is already open.

9.4.3 Binder Affects

Binder Affects... The View ► Binder Affects ► sub-menu contains several choices for altering the standard behaviour to become locked rather than dynamic. For example, if you have a left and right split, ordinarily when you click in the binder what you click on will be loaded in the active split. If Binder Affects Left Editor has been enabled, however, *all* clicks will be sent to the left split no matter which is active. The precise contents of this menu will alter, depending upon whether you are using a vertical or horizontal split orientation.

Alternately, the "Other Editor" choice will cause the binder click to always act in opposition to its normal behaviour, loading the binder click in whatever editor is *not* currently active.

When binder clicks are being diverted, a small indicator will appear in the binder sidebar header, to the left of the title. This iconic depiction will use a darker shade of grey to indicate where future binder clicks will go. This indicator will be removed when operating in the default behaviour, and so its presence is not only useful to determine where a click will go, but that clicks will not be functioning normally in the first place.

You can right-click in this area to directly change which editor the Binder affects.

To reset the binder behaviour to default, use View ► Binder Affects ► Current Editor.

See also:

- Splitting the Editor (section 14.4) for further documentation on how to use editor splits.

9.5 Linking Documents Together

Much like hyperlinks on the Web, Scrivener Links make it easy to create and use a network of text-based cross-references within your project. Since in most cases they do not export from your project, it is safe to use them in the text of your work wherever you need them. They can also be used within Document and Project Notes. Their ability to elevate the visibility of information in your project can turn them into a valuable navigation tool (much like having a personal wiki inside your project), or optionally in providing navigability beyond the project itself, such as cross-referencing that will be useful to the reader.

The creation of links can even be a good brainstorming tool, as you can build your outline with links, while typing. Links appear much like URL links do on the Web. They will sport a blue appearance and be underscored. The link colour can be customised in the Appearance preference tab, under "Customizable Colors", in the Editor section.

The underscore cannot be removed, so even links set to be the same colour as the base text can be identifiable.

To use links, simply click on them with the mouse pointer, as you would in a web browser. Similarly, you can modify how the link is opened using modifier keys. What the modifiers do will depend upon the "Open clicked Scrivener Links in" setting in the Navigation preferences pane (subsection B.6.7). By default Scrivener Links (but not ordinary hyperlinks to the Web) will open a split editor (if necessary) and load the linked item in the split. This will use the ordinary view mode preference, and so is identical to clicking on that item in the binder. If you link to a folder, it will open in Corkboard view, for example.

If you would like to see where the link will take you without actually going there, you can simply hover your mouse over the link for a few seconds, and the full Binder path of the item will be printed in the tooltip.

Right-click on a Scrivener Link to access the "Open Document Link In" sub-menu for additional ways to open a link.

Keyboard shortcuts can also be used to access these optional methods directly:

Other Editor (default)
- Command: open in QuickReference panel
- Option: open in current editor

Current Editor
- Command: open in other editor
- Option: open in QuickReference Panel

QuickReference Panel
- Command: open in other editor
- Option: open in current editor

9.5.1 Creating Scrivener Links

Drag and Drop

- To create a link to a specific item in the binder, drag the item into the text where you would like it to appear, while holding down the Option key when dragging. The blinking cursor will help show you where the final location of the drop will occur. When creating links in this fashion, they will be automatically titled by the name of the document that was dragged.

- It is possible to drag more than one item into the text area at once. Select multiple items in the binder and drag them over as above. Links will be separated one per line.

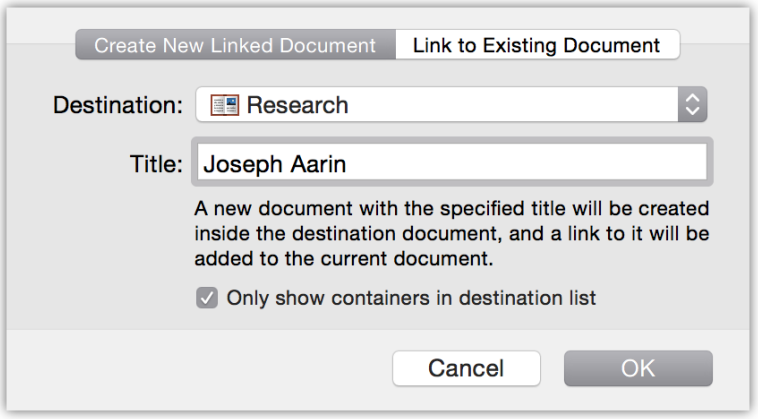
Create New Linked Document
Link to Existing Document
Destination:
Research
Title:
Joseph Aarin
A new document with the specified title will be created inside the destination document, and a link to it will be added to the current document.
Only show containers in destination list
Cancel
OK

Figure 9.3: Create a New Linked Document

- Items can be dragged into the Notes Pane (and the Project Notes window) as well as the main text area, in this fashion.
- In a slight variation of the above, select the text you wish to link to a document and then Option-drag the item onto the selection range. This will link the text to that item without inserting the name of that item.

Select Items and Copy

When you want to create links to items in multiple places at once, it is usually easier to use the **Edit ▶ Copy Special ▶ Copy Documents as Scrivener Links** menu command. Now you can paste (⌘-V) the items into several different places rapidly. As with Drag and Drop, you can select more than one binder item at once, producing quick lists of links. For e-books, this can be useful for creating a custom table of contents, as that system uses Scrivener Links to assemble e-book compliant hyperlinks.

Select Text and Link

Often you will want to link a phrase to a specific document without having the name of that document be a part of the link. To create links in this fashion, select the text you wish to link, and then right-click on the selected text. Use the **Scrivener Link** sub-menu to create a new link. This menu is also located in **Edit ▶ Scrivener Link**.

This sub-menu functions in a very similar manner to the Go To menu (section 9.2), by providing an organised list of every item in the Binder. The menu is organised as follows:

New Link (⌘-L) Brings up a sheet which gives you the option between creating a new item and choosing where to place that item, or to navigate through a list of items

that already exist.

There are two tabs available in the sheet. The default tab, "Create New Linked Document", will let you generate a new document and attach it to an existing container (or optionally, another item, forming a new container).

The "Destination" drop-down menu provides a list of all the containers in the project Binder. Use this to select where the new document should be created. By default, it will select the Research folder for you. If you have unchecked "Only show containers in destination list", then the drop-down menu will display all items in the Binder, allowing you to form new containers by creating new items beneath existing documents[2].

The "Title" field is where you type in the name of the new binder item which will be created. It does not need to match the text of the link.

The "Link to Existing Document" tab will provide a Binder browser displaying all existing items in the project for easy link-up. Using this method is very similar to right-clicking on the text and selecting an item from the Scrivener Link sub-menu, and is mainly provided as a convenience for people using the shortcut to create new links, or wiki linking, discussed below.

Suggestions If the selected text contains material that is found in existing binder item titles, these matching titles will be added to the menu as well as suggestions. This is handy when you have typed out the name of a Binder item, and wish to create a link to it.

This section will not appear if no suggestions are found.

Favorites As with the Go To menu, "Favorites" will be added to the list. Consider adding frequently linked to items to make it easier to access them in the future (section 9.6). This section will not appear if no "Favorites" have been created.

Binder List The remainder of the menu will be organised according to how your project is laid out. Containers will be converted into sub-menus, allowing you to navigate the project and select a link target (much as you would with the "Go To" menu).

Wiki Link Style

An optional method lets you type in new links without using the mouse at all, or even being fully aware of the title you wish to link to. To enable this method, visit the Cor-

[2] If this is unfamiliar, I suggest reading Folders are Files are Folders (subsection 8.1.3).

rections tab of Scrivener's preferences, and turn on "and [[Scrivener links]]" in the Substitutions section.

To use this method, while typing in the text you can enter two double-brackets, type in the title of the item you wish to link to, and then close it with a second pair of brackets, as shown. Scrivener will detect what you are trying to do, and if it finds an exact match to a document, will link it for you automatically. If it does not recognise the text inside the brackets as correlating with an existing title, the New Link sheet[3] will pop-up, giving you the option to either create a new item and place it in the Binder, or via the second tab, "Link to existing document", navigate to an existing document in the Binder and create a non-literal link. Non-literal links are useful when you wish to link phrases in your text without directly referencing the name of the item you are linking to. Once substitution has been performed, the brackets will be removed.

> **Quickly Referencing a Name**: If you don't recall the precise name of the document you wish to link to, a quick way to search for item names without using Project Search (and messing up your current view) is the **Edit ► Find ► Find Synopsis...** tool. Using this window you can rapidly search only titles and text synopsis. Once you know the name of the item you are looking for, close the window and proceed with creating the link.

As with other substitution types, wiki linking[4] works only on newly typed material. If you have previously typed in double-bracketed words, and then enable the option, you will need to re-bracket them.

9.5.2 Removing Links

Links can be removed (including URLs to external web pages) with the **Edit ► Unlink** menu command, without destroying the underlying text. Any link falling within the currently selected text will be removed, so there is no need to be precise about what you select. This is an easy way to wipe out dozens of links at once, something which is of particular use when pasting text from web files, or other Scrivener documents with links you no longer require.

Note that linked comments and footnotes will *not* be deleted with this command.

---

[3] sheet-new_link-new.png

[4] Referring to these links as "wiki links" does not imply that your project can become a wiki. It merely refers to the appearance of the syntax as it is typed in. Scrivener links must always point to an existing resource (even if you create that resource while generating the link) unlike true wiki links, which can be placeholders for items which do not yet exist.

9.5.3 Changing a Link Target

The link target, what Scrivener will display when you click on the link, can be modified in one of the following ways:

- Select the text of the link and use the **Edit ► Scrivener Link ►** sub-menu to select a new target. You can also right-click anywhere in the link text and use the contextual menu to make this selection.
- Select the text of the link in the editor and then and drag the item you wish to link to on top of the selection with the Option key held down.

9.5.4 Updating Link Text Automatically

When using Scrivener Links as a form of visible cross-referencing that will be available to the reader, a mechanism for updating the link text with revised section titles will frequently prove to be useful. Since Scrivener links are themselves merely arbitrary text pointing to another item, it is not possible for Scrivener to determine your ultimate purpose for that link, nor is it possible for it to guess at whether a string of text which was once an exact copy of a title should be updated with the new title, since it is unaware of what that old title was. In other words, if you change a section name from "Absinthe" to "Chartreuse", Scrivener would not be able to discriminate the fact that a link called "Absinthe" pointing at a section named "Chartreuse" is in fact meant to be updated or is any different from a link called "exotic liquors" which also happens to point there but was never meant to be a cross reference. The software doesn't store a history of every name you ever gave to an item, so it cannot guess that "Absinthe" is an old title for that section.

Thus, this action requires an intentional command from you, though the actual updating will be automated. There are two ways to go about fixing links. If you are certain that all of the Scrivener Links you have created in your draft are functioning as cross-references (you have no links at all of the variety which are merely meant to be used for your own reference), then you can select large portions of text at once, even in a Scrivenings session, and fix the links using the **Edit ► Update Links to Use Target Titles** in batches. You may also fix links one-by-one when you come across them with this command (just make sure the whole link is selected).

> **Updating the Link Target**: This feature only updates the visible text of the hyperlink. It does not adjust the target which the link points to internally. If you wish to change the target of the link itself, select the link in the editor, and use the **Edit ► Scrivener Link ►** sub-menu to select a new target, or use the ⌘–L shortcut and access the "Link to Existing Document" tab.

Additionally, if you use a mixture of cross-reference style links and links for your own benefit, you'll probably want to step through them all one-by-one to make sure that the update command only hits those links that are meant to be cross-references. The easiest way to do this is to use the Find by Formatting (section 21.1.6) panel, which contains a handy button you can use to replace the link text with the title of the item it links to. Since the Find by Formatting panel will automatically select link text for you, this will be the easiest way to do spot updates.

9.5.5 Configuring How Links Work in the Interface

Most of the configuration options for links are located within the preferences Navigation tab (section B.6), in the Scrivener Links section. Here you can configure how links will act when they are clicked, and what will happen when new links are created.

Both settings have three options available:

- **Current Editor**: The link will replace the current editing session with the linked item. This method works most like a web browser, and like a web browser you can use the forward and backward buttons to get back to where you were after following a link.
- **Other Editor**: Will use the other split to load the clicked link. If necessary, Scrivener will open up a new split (section 14.4) for you, using the last used split type (horizontal or vertical).
- **QuickReference Panel**: The item will be loaded into a QuickReference panel (section 21.2)[5].

The colour appearance of links can also be adjusted in the Appearance tab (section B.3).

9.5.6 External Links

It is not only possible to link to and from individual items within a project, but between projects as well. In fact, you can copy a special type of URL to the clipboard and then use it to create a link in any program on your computer.

Linking Items Between Projects

The easiest way to link to an item from another project is to use the References feature (section 19.5):

[5] Scrivener Links activated from within a QuickReference panel will always open a new panel.

1. Open the Inspector in the project you want the link to be in and click on the stack of books icon to view the References pane.

2. From the Binder of the project with the item you want to link to, drag that item to the References pane, opened in the prior step.

You will see a link to that item, with the URL filled out. When you double-click the icon on this Reference, to activate the link, the original project will be opened if necessary, and this item will be loaded in the editor.

The URL text can be used to create other links at will. You could for instance drag this Reference line into the main text to create a clickable hyperlink, there. You could also copy and paste it into another program.

Linking From Other Programs

Any software that has a concept of linking should allow you to create useful links back to individual components of your project. This is done via the use of a special type of URL, which can be used to instruct applications on your system to take certain actions. In this case, this link will instruct Scrivener to launch, open the target project and then display the requested resource in the editor. If Scrivener or the project is already open, then clicking the link will have a much more immediate effect.

- To copy an external link, select the item you wish to link to and use the **Edit ▸ Copy Special ▸ Copy Document as External Link**. Now you can paste this URL into any context that takes one.

- Alternatively, right-click on the item directly in the Binder, and select "Copy Document Link" from the contextual menu.

9.5.7 When Scrivener Links can be Compiled

Depending upon the file format used , when compiling your work into a single file, Scrivener Links can be a feature that reside entirely within the realm of Scrivener, being stripped out of the final product, or something that can be used for linking (or referencing) that your readers can make use of. Some examples range from tables of content, cross-references, hyperlinks to the web, and more.

When compiling, unless the target format (such as RTF, PDF and HTML) supports point-to-point linking, they will be stripped from the output, and thus are handy for inserting useful links for your own purposes. For those formats that do support linking, you can opt-out of this behaviour in the Transformations compile option pane (subsection 24.15.2), by selecting "Remove all hyperlinks".

Compiling with Automatically Adjusted Titles

Sometimes you may need a link to adjust the hyperlink text—what your readers will see—to match the given title for the section it refers to. For example, you might need the Table of Contents list to print the chapter numbers, which are also be compiled into the titles. This method requires a few conditions to function properly:

1. Link text which is solely the title of the document they link to.
2. A title prefix or suffix applied to the document level that contains the links. (Additionally, enable the "Update titles in Scrivener links with prefix and suffix settings" option, in Compile under "Title Adjustments".)

This is a more advanced feature of Compile (which is discussed in its own chapter) but in essence it allows you to create a link to document titles in the text which will be altered to match their final appearance in the compiled version. Since titles can be added to, or even entirely replaced, by the compiler, this set of features will ensure that referenced titles will remain valid after export. This can be used even in formats that do not support linking, since it primarily is concerned with keeping the *text* of your document up to date with the final presentation of its structure:

The original text reads:

> To read more about this topic, see How to Grow Better Tomatoes .

The phrase "How to Grow Better Tomatoes" is linked to a binder document of the same name, in the Binder. However, the author has arranged for this title to be prefixed, so that in the final output it will appear as "Chapter 7: How to Grow Better Tomatoes".

With the above options configured correctly, Scrivener will recognise that this link points to a title and that it should be updated to reflect the final version. It will be compiled as the following result:

> To read more about this topic, see Chapter 7: How to Grow Better Tomatoes.

The Table of Contents generator uses this trick to keep its list of documents up to date with the final manuscript version of titles.

> **Pro Tip**: If you link the special placeholder tag, <$p>, to a particular document in the Draft, the compiler will insert a special RTF code which lets compatible word processors cross-reference the actual page number of that item. This technique is also used in the Table of Contents generator, but could also be used for formatting nicer cross-references for your readers.

If you wish to cross-reference a document in a more generic fashion, without displaying the entire title, the <$hn> placeholder can be linked to a source document, if it is numbered likewise. The number of the target section will be printed here, instead of the number for the source document. Note that if the intention is to show the title and the number of the section together, and the placeholder is being added to the title via a compile prefix, there is no need to manually add this yourself. The compiler will update the name to include any prefix, including numbering, for you. This technique will mainly be useful if you intend to cross-reference to a section without including the title, using a generic "(see section 1.2.3)" nomenclature.

E-books are afforded the power to link to items in the Binder which are otherwise not true sections. This can be useful if you wish to link to a particular figure—you could place that figure in a text file all by itself, and then link to that resource with a Scrivener link. This will navigate to the image in most reader hardware and software.

Linking to Images with MultiMarkdown

There is one other case where a Scrivener Link will be used intelligently by the compiler, and that is with the MultiMarkdown format. When a link points to an image file, that file will be included in the output folder when you compile, and the name of the file will substitute the hyperlink text. In this way you can use the MultiMarkdown syntax for images. An example might look like:

![Caption text for the figure](Scrivener Link to Graphic)

Which would compile as:

![Caption text for the figure](name_of_graphic.jpg)

Since the image file, **name_of_graphic.jpg** is exported alongside the source file, it will naturally link up correctly and display in the ultimate output.

See also:

1. Compiling the Draft (chapter 24)
2. Creating a Table of Contents (chapter 23)
3. MultiMarkdown and Images (subsection 22.4.1)

9.6 Using Favourites

Setting binder items as favourites will increase their visibility in a few key areas of the application. One of these has already been covered; the "Go To" menus. Setting an item as a favourite will make it easy to navigate to that item in the future, as it will always be

present at the top level of that menu (with the exception of when the menu is acting as a table of contents for a Scrivenings session).

All of the menus which have favourites added to them:

- Edit ▶ Append Selection to Document ▶ (Also available as a contextual menu when text is right-clicked upon.)
- Edit ▶ Scrivener Link ▶ Creates text links to documents, like URLs in a web browser. (Also available as a contextual menu item.)
- View ▶ Go To ▶ Provides navigation features to any spot in the binder. (Also available from the header bar icon menu; except when operating at a table of contents for a Scrivenings session.)
- Documents ▶ Move ▶ To ▶ Moves selected binder items to other locations in the binder. (Also available as a contextual menu item from the main sidebar.)
- View ▶ QuickReference ▶ Opens a QuickReference Panel for the selected item from that menu.

9.6.1 Adding and Removing Favourites

Adding and removing items from the list of favourites is easy to do. Simply select the item in the binder and right-click on it to view its contextual menu. At the bottom of the menu, there will be two options:

- **Add to Favorites**: Adds the selected document(s) to the favourite list.
- **Remove from Favorites**: Removes the selected document(s) from the favourite list.
- To clear *all* assignments, use the Documents ▶ Favorites ▶ Clear Favorites menu command.

9.6.2 Managing Favourites

Favourites can be managed centrally using the Documents ▶ Favorites ▶ Manage Favorites... menu command. You can drag items from the Binder into this floating window, or use the - button along the bottom to remove selected items that have fallen in disfavour.

Items can be dragged around to change their order in this window. This will also impact the order in which they appear in the various menus where favourites are included.

The items in this list are also valid objects for dragging. This means you can drag items out of this list into header bars to load them, drag them into text editors to paste their contents (or with the Option key held down, to place a link to that item).

9.7 Layouts Panel

The Layouts panel allows you to save different layout set-ups and load them again easily. For instance, in the early stages you might focus on a large corkboard without any other interface to get in the way; as structure starts to emerge, you may have the binder open with the corkboard in the top editor and text in the bottom editor; whilst composing, you may have only the text open, with the binder, inspector and toolbar hidden. By using the Layouts panel to save different layouts, you can switch between these view settings quickly, without having to go through the process of hiding the binder, toggling the toolbar and so forth. You can even specify that outliner column settings and other details will be restored from a layout. To show the Layouts panel, select **Window ► Layout ► Manage Layouts...**, or press ⇧⌘–**0**.

If you do not have any layouts saved yet, you'll be presented with a window and an empty list. You can try the feature out immediately by clicking the **+** button. A screenshot of your current project window will be taken and placed into the right side of the screen, and you'll be given the opportunity to name the layout something meaningful. With a combination of names and screenshots, your layouts should remain accessible and easy to find.

Switching to a saved layout is easy, you can either double-click on the icon next to its name, tap **Enter**, or just select the layout and click the **Use** button in the footer bar of the left list. Another way of switching layouts is to use the main menu instead of the Layout panel. When you save a layout, it will be added to the **Window ► Layouts ►** sub-menu. Since these are assigned a menu command, you can configure custom keyboard shortcuts for quick access to commonly used layouts, using the standard OS X system shortcut tool.

To remove a layout you no longer need, select the layout and click the **-** button in the footer bar. You will be warned that once it is deleted you'll be unable to retrieve it. This warning can be dismissed so that it no longer appears, if you wish.

To update an existing layout with the current project window layout, simply select that layout in the list, and click on the gear menu in the footer bar, choosing **Update Selected Layout**. A new screenshot will be taken and the old settings will be updated to reflect the current window layout.

Layouts can also be exported and installed into Scrivener. Using the gear menu, select the layout you wish to export, and then choose **Export Selected Layout**. The layout will

be saved as an XML file in the location you provide. Use the **Import Layout** menu choice to import a layout you have transferred from another computer or downloaded from the Internet.

Because layouts are separate from projects, there are some things that they cannot save, such as which collection you are viewing, which QuickReference panels are open, or whether the binder has been hoisted to a certain level or switched to viewing a particular collection. Even within one individual project, these details might change or no longer be relevant. Here is a list of things layouts *do* save:

- Binder visible/hidden
- Collection tab interface visible/hidden
- Inspector visible/hidden
- Split type (horizontal/vertical/none)
- Binder mode (whether it affects one or current editors)
- Header bar visible/hidden
- Footer bar visible/hidden
- Editors locked in place
- Ruler and Format Bar visible/hidden
- Selection auto-load in alternate editor settings
- Window size and position and view sizes
- Toolbar visible/hidden and view mode (such as small or large icons)
- Corkboard/outliner/scrivenings/editor mode for each split
- Page View or standard draft view for each split
- Full Screen status [**10.7+ (Lion) Only**]

In addition, optional settings are saved depending upon whether these two checkboxes are enabled for the specific layout.

Save outliner and corkboard settings All corkboard display settings, such as card size, ratio, card wrapping, and so on will be produced when using the layout.

In the outliner, which columns are visible or hidden, and column sorting.

Preserve all meta-data appearance options This will determine whether or not label tinting used in the various areas of the interface; and pin, stamp, and keyword chip visibility in the corkboard, will be produced when using the layout.

9.8 Full Screen Mode

Mac OS X Lion introduces a new feature which make it easy for third-party applications like Scrivener to utilise a special full screen mode which pushes the Dock and main system menu bar off-screen. In addition to reducing these distractions, the application is free to fill every single available pixel on your monitor.

Scrivener's full screen implementation is project window based. This means you can actually run several different projects at once in different full screen spaces on your system. Each project window will feature a full screen expansion icon in its upper-right hand corner. This button corresponds with the **View ▶ Enter Full Screen** menu command[6] and can be toggled with **^⌘–F**.

> **Toolbar Bug**: In the current release of Lion, there is a bug which causes the main toolbar to appear in full screen mode, even if you've set the toolbar to be otherwise hidden. Fortunately it will be auto-hidden along with the main menu, and curiously if you select **View ▶ Hide Toolbar** while in Full Screen, it will then act as it should (but it will not stay that way permanently).

Since Lion creates a new "Space", dedicated for any project window that has been set to use full screen, any peripheral windows that had been opened (such as the colour palette, Find tool, QuickReference panels, etc.) will not be conveyed to that new space for you. New windows that you open while working in full screen will remain in that view, but existing windows will not, and cannot be moved afterwards without closing them and then re-opening them.

> **Full Screen with Multiple Displays**: Due to how Mac OS X Lion handles full screen mode, if you are using multiple displays and enter full screen mode, all of the other displays will be blacked out. This is unfortunately out of our control.

9.8.1 Slide-out Sidebars

[**10.7+ (Lion) Only**] On smaller screens, such as found on the 11" MacBook Air, you might want to dismiss both the binder and inspector sidebars automatically to maximise

[6] In prior versions of Scrivener, this menu command would enable the distraction free writing mode. Due to naming conflicts, this feature has been renamed to Composition Mode (chapter 17).

the amount of space provided to the editor(s) or to better focus on the content. The "Hide binder and inspector when entering full screen mode" option, found in the General preference tab (subsection B.3.5), will do this for you.

When hidden, sidebars will slide in and out when moving the mouse to the relevant side of the screen—or if you use any of the keyboard shortcuts that might otherwise reveal these sidebars. For example, pressing ⌥⌘–R or using **View ▸ Reveal in Binder** will typically show the binder if it has been hidden, and so in full screen it will slide out the binder so you can see the position of the file you are editing.

9.8.2 Full Screen and Layouts

Saved Layouts (section 9.7) will remember if they have been saved in full screen mode. Switching to a Layout that has been marked with a full screen icon beside it will enter full screen as well as re-arrange the window, making for a convenient way to work with special-purpose Layouts intended to use a lot of space. If you do not wish for a Layout to transition you to full screen, you can hold down the Option key when selecting the Layout for use.

You may also select up to one full screen layout to use as a default. This will cause Scrivener to trigger the use of this layout even when using the standard menu command, rather than using the Layout features to access it. You can think of it as a way of setting up Scrivener to have a different form of interface when entering full screen.

If no Layout has been selected as a full screen default, then the current window settings will merely be expanded until the project window fills the screen.

9.8.3 Full Screen and Fixed Width Editing

Most of the default preferences have been chosen to complement the full screen feature. When you enter full screen, the editor will switch automatically to using a fixed width display, rather than a fluid width as it does normally. This means that no matter how wide your screen is, the text editor will keep itself to a readable line length. If you'd prefer for your editor to print text across the entire display no matter what, then you'll want to disable the "Use fixed width" and "Full screen only" options, in the Editor preferences tab (section B.7).

9.9 Filtering Items in Views

One of the ways in which you can work in Scrivener is with interleaved notation amongst your work in progress. This is a method that some prefer, as it keeps their thoughts and ideas immediately contextual with the text itself, and it supports a working method where notes are more extensive than what can easily be put into annotations attached to

the text itself. In Scrivener, you can place text documents right alongside the documents used to contain portions of your Draft, and then instruct these special note documents to not compile (subsection 10.1.5)—in other words when you turn the Draft folder into one long file, these note files will be hidden.

Ordinarily, when you view a portion of the draft with the Scrivenings view mode (section 15.9), you will see all of your notes and draft text together. This can be advantageous, especially if you use an alternate font for your note files—but there will be times when you want to just see the compilable text that will become your book.

You can also select one or more containers and use the **Documents ▶ Open ▶ (with) Compilable Subdocuments** menu item. The name of this menu item will change depending on whether or not you've set up your preferences to consider the container as part of a Scrivenings session. Some prefer to omit the container, and so set "Include folder text in Scrivenings (composite text mode)" in the Navigation preferences tab (subsection B.6.1) to off. When this setting is disabled, the menu item will omit the 'with', and only the children of the selected container will be included in the flat list.

A quick way of doing this is to hold down the Option key and click on a container containing the portion of the draft you wish to view in this fashion. For some people who use minimal organisation, this might mean using the Draft item itself. The result of this action will be a flat multiple selection of items. You can actually view these in any group view mode that you prefer, though in most cases you'll probably want to use the Scrivenings view mode.

Chapter 10

Setting Up Meta-Data

Documents of any type in Scrivener can have various meta-data associated with them. Meta-data is a way of talking about something without changing it directly. A simple example from the analogue world could be a Post-It note on a paper-clipped stack of paper. The Post-It note is a kind of meta-data, as is the paper-clip.

In Scrivener, meta-data can be applied in similarly analogue fashion, like stamps across the face of an index card, or what colour of pin or marking it has in its corner. Other types of meta-data in Scrivener are more abstract, and you can even make up your own if the built-in types are not enough.

Some of this meta-data can be viewed in the corkboard, the QuickReference panes and outliner views, but the main interface for viewing and editing all of the meta-data for a given document is the Inspector (chapter 19). This section will brief you on the various types of meta-data available, and how to create more types if necessary, it will not go into all of the details of the Inspector pane; please read that section to gain a full understanding of how meta-data is assigned to documents.

10.1 Meta-Data Types

The basic meta-data types include the title and a text or image synopsis. The title is the most important component of any item in the binder, as it will be used to identify the document in the many views, menus, and export methods. When exporting as files, filenames will be set according to the title field; when compiling, titles can be used to generate headers for that section; index cards use the title; titles may also be used intelligently by the compiler when linked to the document by that same name; and the outliner is composed of a list of titles.

The synopsis is a plain-text field, primarily intended to be a short summation of the contents of the document, though how you choose to use it is entirely up to you. The synopsis is displayed in a variety of ways.

1. Corkboard: used to display the content area of the card.
2. Outliner: will be placed beneath the title (which appears in bold), by default.

3. Binder: revealed in the tooltip when hovering the mouse over the corresponding title.

Instead of the text synopsis, you can elect to use an image to represent a document on the corkboard. This image will be used on the corkboard and in the inspector, but whatever text exists in the standard synopsis field will be used in the outliner and binder as described above, and in the various export and print options that include a synopsis field. The image is not a *replacement* for the synopsis, but an override for the corkboard.

In all cases, specific types of meta-data can be displayed as columns in the Outliner view, including custom meta-data.

10.1.1 Setting Up Label & Status

The next two forms of meta-data are flexible in what you can refer to them as. Using the **Project ► Meta-Data Settings...** (⌥⌘–,) you can set up custom names for both the label and status, and the interface will adjust accordingly, no longer referring to them as "Label" and "Status" for that project. If you wanted, you could have "POV" and "Location" instead of "Label" and "Status", or "Focus" and "Type", or "Monkeys" and "Bananas", or anything else you want. The documentation will continue to refer to them as labels and status for convenience. They are also, after the title, the most visible items in the entire interface. They can be represented in the corkboard and outliner views, are best administered in the inspector, are given a priority position in the QuickReference panel, are displayed in search results, and are given priority placement in most of the printing methods. Labels, having an associated colour, can also be used to visibly tint various interface elements, such as binder or outliner rows or even the entire index card "paper".

The default blank project and many of the templates come packaged with a few example status & labels, but you will most likely wish to create your own. To add or remove project labels, use the **Project ► Meta-Data Settings** menu item. The interface provides tabbed tables for four meta-data types.

Label and status lines can be added or removed with the **+** and **-** buttons. Edit an existing entry by double clicking on the text. For labels, the colour can be changed by double-clicking on the colour chip to the left. The order in which these items appear in their respective menus is determined by their arrangement in this table. Click and drag rows to rearrange them.

When you create new items in the binder, either by adding items or importing, they will automatically be supplied by a default label and status value. Ordinarily this will be assigned to the "No Label" or "No Status" choices, which is a special entry that cannot be deleted, but they can be renamed. If you wish to change this behaviour, select the alternate value by clicking on its row, and then click the **Make Default** button.

Once you click the OK button, the interface will update with your changes it necessary, and the new options will be available from respective meta-data drop-down menus throughout the interface.

10.1.2 Label Colours

Labels can have a colour associated with them and are completely customisable—you can even rename "Label" to "POV" or something that better fits your project (see below). Labels are one of the most flexible meta-data types in terms of how visible they are in the interface, so you can pick a colour to reflect how prominent you wish them to be (richer more vibrant colours will stand out amongst pastel choices) depending on their associated material's content. Label colour can be expressed in the following ways, nearly all of which are optional, and can be toggled in the View ▶ Use Label Color In ▶ sub-menu.

1. Corkboard: will be used to shade the upper right corner of the card when using the rounded card theme. When using the classic card themes, the pin will be coloured according to the label. Use View ▶ Corkboard ▶ Show Pins or ^⌘–P to toggle the visibility of pins. The visual style can be changed in the Corkboard preference tab.

2. Outliner: can be set to tint the background colour for each row using the label colour with View ▶ Use Label Color In ▶ Outliner Rows.

3. QuickReference Panels: the label is one of the priority meta-data values always visible at the top of the QuickReference panel for each document.

4. Binder: can be set to highlight the background of each binder item using the label colour in a manner similar to the Finder. Toggle this behaviour with View ▶ Use Label Color In ▶ Binder.

5. Icons: enabled with View ▶ Use Label Color In ▶ Icons, this option will tint document icons wherever they appear.

6. Index Cards: enable with View ▶ Use Label Color In ▶ Index Cards this optional method will tint the "paper" colour of index cards wherever they appear.

The label can also be used to tint colour printouts when printing corkboards and outliners.

10.1.3 Status Stamps

As with the label, the status field can have its representative name altered to suit your project's unique requirements. By default, this field represents the status of a document in terms of its completion, such as "To do" or "Rough draft", but this field can be used

for whatever purpose you desire. Unlike the label, there is no corresponding colour, and so its display potential is more limited.

Corkboard: can be displayed as a diagonal "stamp" across the face of the card. The appearance of this can be adjusted in the Corkboard and Appearance preference tabs, and display can be toggled with View ▶ Corkboard Options ▶ Show (Hide) Stamps or ^⌘–S.

QuickReference Panels: the status is one of the priority meta-data values always visible at the top of the QuickReference panel, for each document.

10.1.4 Using Keywords

Each document can have a list of keywords associated with it. These are useful for making documents easily searchable—for instance, you can list all characters and locations connected with a scene in the keywords even if they are not mentioned explicitly in the text. Creative uses for keywords also include extended status control, plot management, and whatever else you can think of.

Like labels, keywords have an associated colour which will be displayed beside the keyword in the inspector list, as an underscore in the relevant outliner column, and in the Project Keywords window (see below). Additionally, keywords can be used in the corkboard as small coloured squares along the right-hand edge of the card. The visibility of these can be toggled with View ▶ Corkboard Options ▶ Show (Hide) Keyword Colors, or ^⌘–K. The display preferences for these can be adjusted via the corkboard display controls (subsection 12.1.5).

When exporting or printing, keywords will be displayed on a single line, separated by commas.

10.1.5 General Meta-Data

In addition to the primary types of meta-data, there are a few smaller items, located in the General meta-data section of the Inspector:

Modified The date and time the document was last changed and saved.

Created The date and time the document was first created. You'll need to click on the "Modified" label, beside the double-arrows, to switch to see this data.

Include in Compile A checkbox determining whether the document should be included in the draft when exported or printed via Compile from the File menu. If checked, the document is exported or printed; if not, it is excluded from the export/print (this can be temporarily overridden in Compile). This way, you can have documents in the Draft folder that act as notes that are never included in the final draft. Note that this setting really only has any meaning for documents that

are inside the Draft folder, though it can be set anywhere, allowing you to modify text documents outside of the Draft in cases where they might at some point end up being compiled, or used as templates for files that will be compiled. The checkbox never has any meaning for media entries, as these items can never be compiled.

Page Break Before If this checkbox is ticked, a page break will be inserted before the document when compiled. Useful, for instance, if the document marks the beginning of a chapter. As with Include in Compile, this only has any meaning for text documents inside the Draft folder.

Compile As-Is All options set in the compile formatting options pane (section 24.11) will be ignored for the checked document. This means no extra material will be added, nor will text be removed, including titles, and no reformatting of the text will be done (the one exception to this being when a Quick Font Override (section 24.26) is in use). As with the above two items, this checkbox has no meaning for items outside of the Draft.

10.1.6 Custom Meta-Data

Custom Meta-data is a free-form feature which you may find a use for if you are looking for the ability to mark sections with a specific piece of information that is often different for each item in the outline, but has a common theme. This is in contrast to things which have common information shared amongst many items—Labels or Status might do better for that—or things which lack a cohesive theme where Keywords might be best.

An example of something for a work of fiction with a cohesive theme and a lot of variety in what gets stored for each item could be timeline information. It is very likely that most of the items in the outline will be taking place at different points of time. Thus a special column for timeline date with a free-form field is an exemplary usage for custom meta-data. Another example, from a non-fiction book on edible plants, could be Latin names for the plant discussed in that outline item. This information is something you would probably want to print in the book as well, but it might be useful to you to have a concise list of Latin names in the outliner as a dedicated column.

And that brings us to presentation. Of all the meta-data types, custom meta-data are perhaps the least visible. In fact, there are only two ways you can view them: in an outliner column or in the dedicated inspector pane. As each field can be added to the outliner as a column, that means you can sort by their values. This can prove quite useful for some types of information. In the Inspector, you will be presented with a linear list of text fields, each headed by the name of the field. This presentation makes it easy to write as much as you need into the fields, or quickly see at a glance all of the details of the item you are viewing.

Hopefully you get the idea. You can of course use them however you like, just keep in mind that they will not auto-complete (like keywords) or provide you with a list of stock answers (like labels or status will). Custom fields are fundamentally free-form plain-text fields, much like the Synopsis.

If you're still foggy on how this might be useful to you, a good example of custom meta-data in use is provided with the application. Try creating a new project using the Recipe Collection template in the Miscellaneous category. Don't worry, you can toss it later if you don't need it, but take a moment to look at how custom meta-data has been used to amplify the information storage techniques available to the project.

Custom-meta usage is generally limited to the inspector and outliner, though it can also be exported along with the rest of the meta-data when compiling, exporting binder items, and printing. The text content of custom fields can be search, as a unified group, using Project Search (subsection 21.1.2).

See Also:

- Custom Meta-Data Pane (section 19.7) for information on using meta-data in the Inspector.
- Managing Columns (subsection 12.2.1) for details on how to add columns to your Outliner view.

Managing Custom Meta-Data

Custom field management is done in the "Custom Meta-Data" tab, within the **Project ▸ Meta-Data Settings...** window. For each column you wish to add, press the + button (Figure 10.1) and supply a field title (which will be displayed as a header for each text block in the respective Inspector pane, or a column header in the outliner). Each field has two options:

Wrap Text Useful for fields where you want to display all of the entered text, even if that text causes the row to expand in height in the outliner. Unwrapped columns will only ever show the first line of text unless you edit the field. This holds true for the inspector pane as well, unless you choose to wrap all fields regardless of their settings.

Colored Text When checked, the colour selection tool at the bottom will be enabled, allowing you to customise the text colour for this column. Text colour will be displayed in the inspector (for the title of the field, not the text body) and outliner columns for the text content of each row.

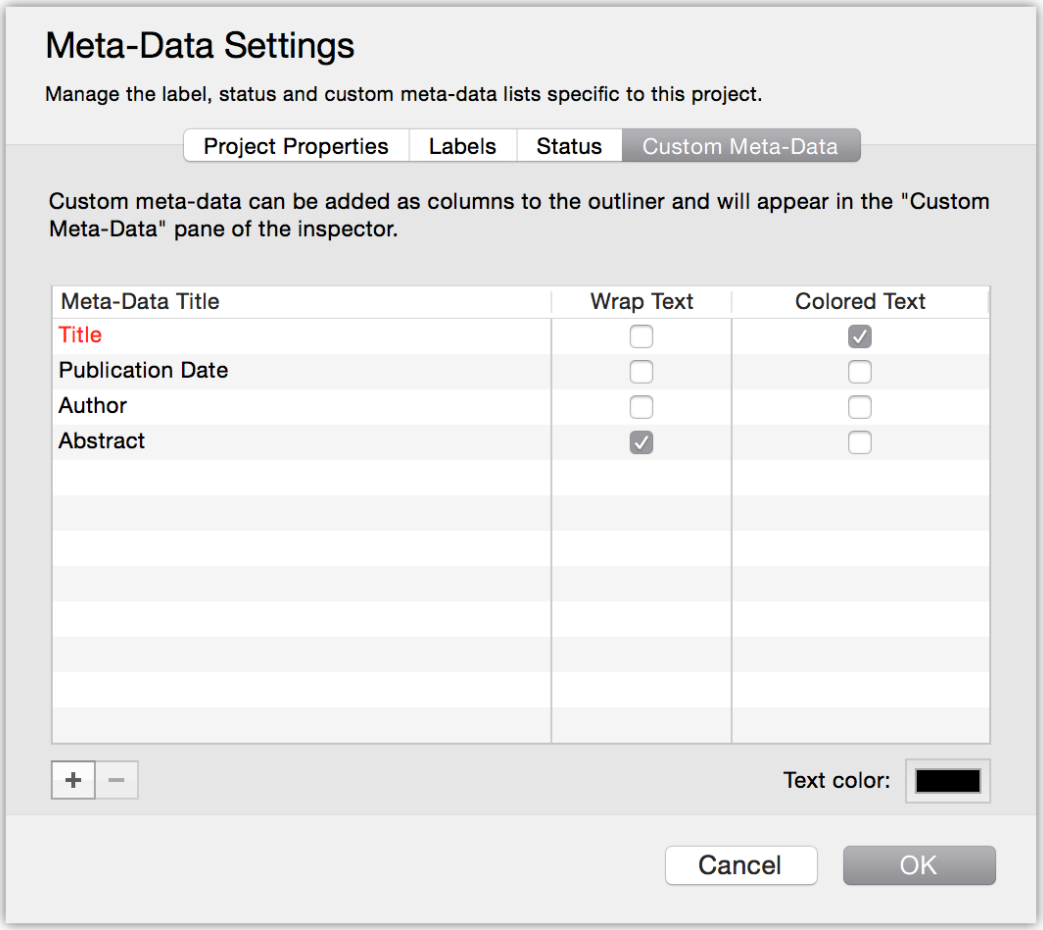

Figure 10.1: Project Custom Meta-Data Setup Tab.

Changing Appearance Order

As with label and status, fields can be reorganised by dragging rows in the configuration table. The order displayed here will impact the following displays:

- Their order within the Inspector sidebar panel.
- Their appearance in the Outliner column view menu.
- Most importantly, the order they will be printed when compiling with meta-data enabled, as well as within the special meta-data text files that can optionally be included when exporting files.

10.1.7 Exporting Meta-data

All meta-data can be exported in a variety of ways, usually text-based for maximum compatibility, so you needn't fear having important organisational information getting locked-in with the project format. Meta-data can be exported in the following fashions:

- Compiling: when compiling the Draft, enabling meta-data export is an optional feature of the Formatting pane. The various types of meta-data will be exported into the Draft in a variety of ways best suited to the type of data involved.
- File export: when exporting files from the Binder, meta-data can be placed into optional "sidecar" plain-text files corresponding to the main document being exported. This method is used to preserve the integrity of the original document, and to enable meta-data export in formats which would not otherwise allow text data to be inserted into it, such as mp3 files.
- Drag and Drop: When dragging documents from one Scrivener project to another, all meta-data will be preserved where applicable. In some cases, like labels and status, these values will need to already exist in the target project.
- Outline and Corkboard printing: both of these printing methods optionally allow meta-data to be expressed in the printout. Unlike the above two examples, these allow for a more fine-grained approach, letting you export just the label or keywords if you want.

10.2 Copying Meta-Data Between Projects

Meta-data can be easily copied between projects by opening both projects at once, revealing their respective **Project ► Meta-Data Settings...** panels, and then using drag and drop to move selected labels, status or custom meta-data fields from one to the other. To copy keywords from one project to another, use the same technique with both respective **Project ► Show Project Keywords** windows visible. Whole nested groups of keywords can be copied at once.

In addition to this, dragging items from one project binder to another with assigned meta-data will also update the target project if it does not yet have those pieces of meta-data set up within it. This also includes keywords.

> **Changing the Defaults**: If you find yourself doing this often, you might consider making yourself a custom project starter template (subsection 7.10.2) with common meta-data values already configured to your liking.

10.3 Project Keywords

The Project Keywords window holds all of the keywords in use by the project. As you enter new keywords into the Inspector (section 19.6), they will be added automatically to the central project list. You can access this pane via **Project ▶ Show Project Keywords** (⇧⌥⌘–**H**) or by double-clicking any of the keywords in the Inspector pane.

You can create or delete keywords by using the buttons at the bottom of the panel:

- The left-most button creates a new keyword as a sibling of the selected keyword
- The second button creates a new keyword as a child of the selection.
- The third button deletes the selected keyword(s).

Keywords are automatically sorted alphabetically. You can drag any keyword onto any other keyword to make the dragged keyword a child of the one onto which it was dropped. For the most part, the hierarchical structure of keywords are really only an organisational convenience with no effect on how the keywords are represented in documents themselves. There is one exception to this, below.

To assign keywords in the panel to a document in the main window, select and drag the desired keywords from the panel and drop them into the keywords table in the inspector of the main window, the header view above the document editor or onto the document in the binder, corkboard or outliner. If there is a multiple selection in the binder, corkboard or outliner and the keywords are dropped onto one of the selected documents, the keywords will be applied to all selected documents. Hold the Option key down whilst dragging to drag not only the keyword but also any parent keyword(s) under which it is grouped.

You don't have to use the keyword panel to create keywords, of course—you can also enter keywords directly into the keywords table, in the inspector (section 19.6), to apply them to the current document. Note that whenever you apply keywords via the inspector, they will automatically be added to the list in the panel if they are not already contained there, so that the panel should always have a complete list of the keywords you are using in your project no matter how you have created them.

Keywords can be managed from the panel. By changing the name of a keyword in the project keywords window, all of the associated keywords in the project can be updated in the background. When removing keywords from the panel, they will also be removed from all of the documents they had been assigned to. To change the colour that represents the keyword, double-click on the colour chip to the right of the keyword.

The project keywords window also provides an easy way to search for keyword usage in your project. Simply click on the keyword you wish to look for, and click the **Search**

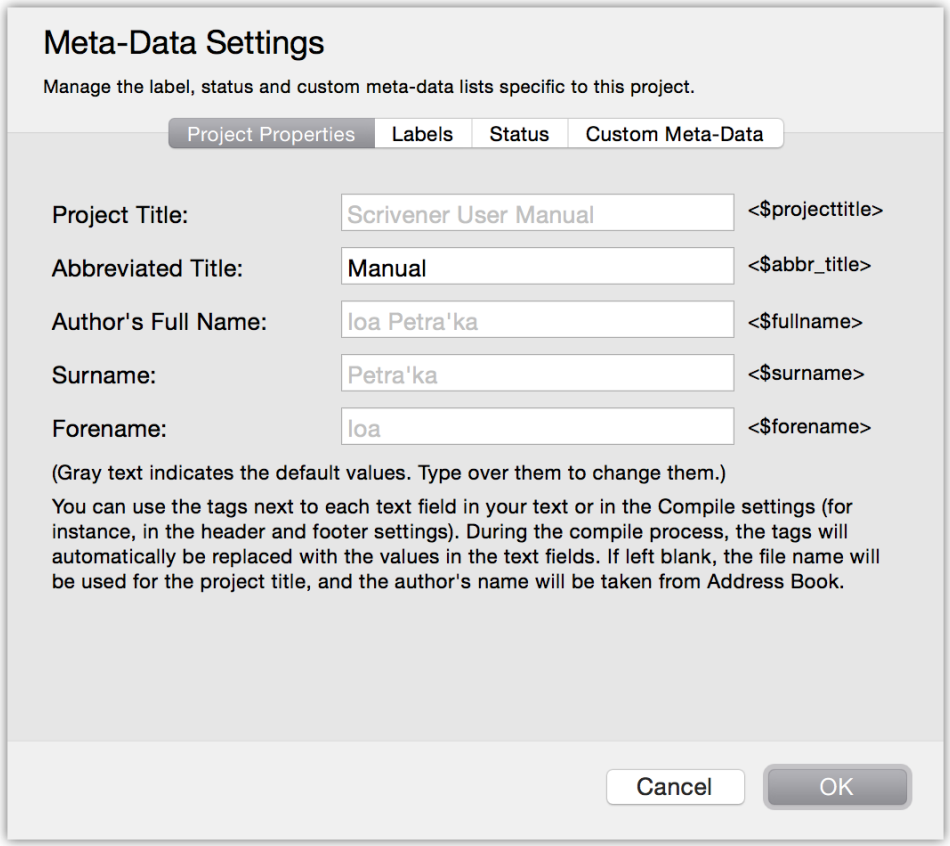
Meta-Data Settings
Manage the label, status and custom meta-data lists specific to this project.
Project Properties
Labels
Status
Custom Meta-Data
Project Title:
Scrivener User Manual
<$projecttitle>
Abbreviated Title:
Manual
<$abbr_title>
Author's Full Name:
Ioa Petra'ka
<$fullname>
Surname:
Petra'ka
<$surname>
Forename:
Ioa
<$forename>
(Gray text indicates the default values. Type over them to change them.)
You can use the tags next to each text field in your text or in the Compile settings (for instance, in the header and footer settings). During the compile process, the tags will automatically be replaced with the values in the text fields. If left blank, the file name will be used for the project title, and the author's name will be taken from Address Book.
Cancel
OK

Figure 10.2: Project PropeSetting Up Meta-Datarties Tab

button in the bottom of the panel. If you instead wish to search all text in the project for that keyword (instead of just real keywords), hold down the Option key and click the **Search All** button.

You can also search for combinations (using the Boolean AND logic) of keywords by selecting more than one keyword and then clicking **Search**. All documents containing *all* selected keywords will be returned. When searching for only one keyword, the "Exact Phrase" method will be used.

See Also:

- Project Search (subsection 21.1.2)
- Keywords Pane (section 19.6)

10.4 Project Properties

The fourth tab of the **Project ► Meta-Data Settings...** panel contains some optional labels that will be used in some of the built-in compile format presets, and can be used on your own using the referenced placeholder tag to the right of each field (which can be copied and pasted for your convenience).

As you might note from looking at this panel, Scrivener will attempt to use reasonable defaults for these fields if they are not changed by you.

Project Title The Project Title is ordinarily taken from the name of your project file on the disk, but in cases where that does not match reality (as it is often convenient to use naming conventions in your project file names to set large revisions apart), you can use this field to override the default and use your own title for the work.

Abbreviated Title Especially useful in headers, long titles often need to be shortened to fit in the header area, so you can optionally use the <$abbr_projecttitle> placeholder to display a shortened version of it. If nothing is supplied here, it will use the project title value (modified or not) as a fallback.

Author's Full Name, Surname, and Forename Ordinarily, these three fields will pull information from your Mac's Contacts.app (which may be called "Address Book" in older versions of the operating system). program, using the default card set up by your system when you first set up your computer. If you need to use a pen name, these would be the fields you should modify. Do note that if you started with a built-in template, some of these values will already be evaluated to text, particularly in any title page material. Since they have been evaluated to text and are not using a token, you will need to edit them separately; this panel will not update them.

> **Pro Tip**: If you would prefer Scrivener to always use another identity than your Mac's main contact card, and cannot change that card for other reasons, you may tell Scrivener to use another card instead. Simply type "(Scrivener:UseMe)" into the note area of the contact card you wish to use from this point on for filling in all name and contact information values. This change should take effect immediately, as soon as the contact card is saved. However it will not change any existing text that has been generated using template tokens in the past.

Chapter 11

Gathering Material

11.1 File Import

Whether you are using Scrivener for the first time and want to use it with documents you have already created in other programs, or whether you just have reference files lying around that you want to bring into an existing project, the chances are that at some point you will need to import documents created in other word processors and programs into your Scrivener project. Fortunately, this is very easy.

Importing files into the project binder means that these files will be copied (and possibly transformed into a fashion that Scrivener can use better) into the project itself. The original copies on your disk will not be removed or altered in any way. While it is possible to easily edit many file types from within Scrivener with an external editor, if you wish to retain maximum external control over the files, you might wish to use References (section 19.5) or aliases (subsection 11.1.3) instead, which are useful when your research material measures in gigabytes.

There are two ways to import documents from other programs:

1. Drag and Drop. In the Finder, select the files you wish to import and then just drag them straight into the binder (not the corkboard or outliner) in Scrivener. When dragging folders, all of the contents of that folder will be added recursively, and the file structure on your disk will be recreated in the binder.

2. The File ▶ Import sub-menu provides some handy methods for bringing existing material into your project binder, including directly off of the Internet, if you have the URL. In cases where the imported material has some sort of innate or optionally defined structure, Scrivener will attempt to convert that into outline hierarchy; documents with split markings can be set up to create many documents upon import; Final Draft files can be split by element; and MultiMarkdown files with their innate header structure will be converted to outline hierarchy.

The first time you import anything, you will be presented with an informative window describing important information pertaining to the various limitations of the import

feature. If you do not wish to view this warning every time you import files, be sure to check “Do not show this warning again”.

When importing text documents, they are internally converted to the RTF format so that Scrivener can work with them easily (again, note that this has no effect on the original file on your disk, only on the copy that is made inside the Scrivener project during the import process). This can cause some loss of data for some file types, or documents with complex embedded data, like Excel spreadsheets.

When using the import menu items, material will be imported according to where the current binder selection is set. If you have a line selected in the binder, the material will be inserted between the selected line and the one below it. If more than one item is selected, the last item (in top-down fashion) will be used as the insertion point. This means that some forms of import will be disabled depending on where your selection is. If your selection is in the Draft, you will be unable to import media of any kind; if your selection is in the Trash, all importing will be disabled. The only exception to this is importing another Scrivener project, and importing via the Dock or when using the PDF print service. These ignore binder selection and create top-level items and/or folders at the bottom of your binder.

When using drag and drop, the dropped material will be placed wherever the insertion indicator in the binder shows. If you try to drag media into the Draft, the drop will be prohibited until you move the mouse out of that folder.

11.1.1 Supported File Formats

Scrivener supports the following text types for importing:

- RTFD (rich text format directory): A proprietary Apple rich text format commonly used by Mac Cocoa applications.
- RTF (rich text format): the universal rich text standard. This is often the best format to use for importing from word processors, purely upon the basis of speed. With the new improved converters you can now import high-quality .doc/x and .odt files into Scrivener directly, but with very large batches of documents you may find this method to be too slow (although batch converting a hundred documents to RTF can also take a long time without a good tool for doing so).
- DOC & DOCX (Microsoft Word format): if you intend to import .doc/x files directly, you should use the improved converters for doing so (you’ll be asked the first time you try). This method is slightly slower, but the results will be much better than using the OS X converters. DOCX is only support on Leopard and greater with improved converters disabled.

- ODT (Open Document Text): the same statement above applies to OpenOffice .odt files. ODT is only support on Leopard and greater without the improved converters.
- TXT (plain text): note that Scrivener tries to import all plain text files using Unicode UTF8 encoding; this should be absolutely fine in most cases, but if a plain text document gets imported as gibberish you may need to convert it to UTF8 format using TextEdit before importing it into Scrivener. If all else fails, use copy and paste. Files with the following extensions will be imported as text as well: .log, .do, .xml, .tex, .mmd, .md and .markdown. Additional extensions can be added to the Plain Text Import Types (subsection B.10.2) tab of the Import/Export preferences pane.
- PDF (portable document format): standard format for preserving and publishing documents in a read-only format.
- HTML (hypertext markup language): the language of the Web. Imported HTML files can be converted to either rich text to extract just the content, or to WebArchive for full page archival. WebArchive files themselves can also be imported directly into Scrivener.
- FDX (Final Draft format): using the standard document format for Final Draft, you can import scripts directly into any area of the Binder and have those imported documents converted to Scrivener's script formatting. Better yet, use the File ► Import ► Import and Split... command to automatically split an FDX file into smaller documents, with the ability to select where splits occur.
- .fountain (Fountain plain-text screenplay): like Markdown, this format is more of a convention of how you type within a plain-text file. This makes the format suitable for editing on nearly any digital device. Screenplays written in Fountain can be dropped straight into the binder, or automatically split by scene.
- FCF (Final Draft 5–7 File Converter Format): a special format for conveying scripts into and out of Final Draft. Use this format to import scripts into Scrivener if Final Draft 8 or greater is not available. As with FDX, can be automatically split into individual files.
- OPML (Outline Processor Markup Language): commonly exported from outlining style applications, this format well let you transfer an outline tree from one application to another. Many applications will also save text notes attached to outline nodes, and if Scrivener detects them, it will use this material in the main text editing area. This behaviour can be changed in the Import & Export preference

tab (section B.10). By default, each top-level outline node will be imported into the binder with all of its children. To contain the entire import in a single folder, you'll find an option in the aforementioned preference tab.

- .indexcard (Index Card for iPad files): if you have no need for keeping Index Card and Scrivener synchronised (section 13.4), and just want to import some cards you've created, you can import an .indexcard file straight into the Binder. This will work in a fashion similar to OPML. It will create a sequence of matching cards for you based on the original cards in Index Card. This feature will not organise cards from "stacks" into folders.

- No extension. Documents with no extension get imported into Scrivener as plain text files (note that this can often be a source of confusion - if you try to import an RTF or DOC file that has no extension, when you import it into Scrivener you will see all of the raw code because it will be imported as plain text. Make sure you add the appropriate extension before importing to ensure that Scrivener recognises it as an word processor file).

As well as these text file types, Scrivener also supports all of the main image file types (TIF, JPG, GIF, PNG, BMP etc), all of the main QuickTime audio/visual formats (MOV, MPG, WAV, MP3 etc), PDF files, HTML, and .webarchive file types for importing saved web pages from Safari and other Mac applications that support it.

Unsupported Formats on Snow Leopard and earlier: Any file type can be imported, even if it is not supported by Scrivener. Such files will appear in Scrivener's editor as an icon with a link which, when clicked on, will open the file in the program associated with it. Or you can click on the Quick Look icon in Scrivener's footer bar to view the file in a Quick Look panel. Unsupported files in the binder, while they cannot be viewed in a split or edited are otherwise fully capable of everything other media files are capable of. They can be organised, keyworded, annotated, and referenced. "Import supported file types only" must be disabled in the Import & Export preference tab.

Unsupported formats on Lion and later: Most of the above still applies. The only difference will be that Quick Look data is loaded into the split view, instead of simply revealing an icon. If the selected item does not have any Quick Look data, then the icon view will be used instead.

See Viewing Unsupported Document Types (subsection 14.3.5) for more information on how to work with these types of files.

11.1.2 Web Page

This lets you enter the URL of a web page that you would like to import into Scrivener. The web page will be archived on import, meaning you will no longer need to be connected to the Internet to view it (you can convert it to a text file afterwards by using **Documents ► Convert ► Web Archive to Text** should you wish to edit the file). Web pages cannot be imported into the Draft (which only supports text), so you must ensure that a non-Draft item in the binder is selected for this item to be available, or change your preferences to import web pages as rich text files. When such conversion is performed, Scrivener will store the page's original URL in References (section 19.5).

> **Functional Web Pages**: Many web pages these days are "functional" in that you can do things inside the web page after you load it. A good example of this is Gmail, Google Docs, or even a simple search form. These sorts of pages, if they require a login, will not import correctly. You will need to use copy and paste, or somehow export the material from the web site to your drive in order to archive them.

You may also drag and drop URLs from many supporting browsers' location bar, into the Binder itself, just as though you were importing a file from your hard drive.

11.1.3 Linking to Research Material

The default method for bringing research material into your project results in the original file being duplicated into your project. From there, it becomes a part of the project in every way. If you move the project to another computer, your research material will follow. There are a few downsides to this. One is when you need to continue refining and editing these external resources. While this can be done from within Scrivener, by loading the file in an external editor—if you want these files to be up to date and available to other applications, fully importing them represents a compromise. Another downside is project size. While the project format itself is capable of great quantities of imported material, this can slow down the automatic backup routine, and will make creating your own backups less convenient, especially if the project contains many gigabytes of research material.

The solution is to link to these items instead of importing them. The special menu command, **File ► Import ► Research Files as Aliases**, can be used to establish a link between the original file(s) and your project. Linked resources will be displayed with a small arrow in the lower-left corner of the icon, in Scrivener. If these files are renamed or moved, the link will adjust accordingly.

However if you move the project to another computer, the research files will no longer function. The links will stay in place, and all meta-data or organisation you have assigned

to them will remain, but the source of the file will be offline until you return to the primary computer where the link was established.

This feature is only available for non-text research (PDF, multimedia, and web files) that Scrivener can display in the editor window. On account of technical limitations, it is not possible to link to a supported word processor file or plain-text document.

11.1.4 MultiMarkdown File

This imports a MultiMarkdown text file into the project by splitting it into multiple files based on its structure, and recreating that structure in the binder. The Meta-data block will be inserted as the first document at the top level, which Compile will use as auxiliary meta-data on top of any Compile Meta-data settings for MultiMarkdown. Below that, each header found in the MMD file will be used to create a new document at an appropriate level of depth, with any text following that header up to the point of the next header included in that document.

11.1.5 Plain Text Formatted Screenplay

Not to be confused with the Fountain format. When importing screenplays from Movie Magic Screenwriter, Celtx, and other programs that export plain-text scripts, you can use this import utility to have them converted into Scrivener's script formatting automatically. You can optionally choose to have the imported file split into multiple scene documents for you. If you are working in Final Draft, you should use the FDX format instead, with the **Import and Split** utility, or FCF for a version of Final Draft prior to 8.

11.1.6 Scrivener Project

Allows you to import another Scrivener project into the binder of the current project. The files of the other project will be imported into a folder entitled "Imported Project (original project name)" at the bottom of the binder, with the full binder structure intact. Keywords, project references and any project notes are imported as text document(s). Project notes will be imported as files into a "Project Notes" folder, placed at the top of the main import folder. This command can also be used to import, and therefore restore, corrupted projects. If you have a project that has somehow become corrupted so that it can no longer be opened in Scrivener, use **File ► Import ► Scrivener Project...** to have Scrivener do its best to retrieve all the data. All the text, synopses and notes will be retrieved, along with snapshots. The outline structure may not be recovered from badly corrupted projects, but every attempt to import the raw data will be made.

11.1.7 Import and Split

There are three modes for this tool:

1. Plain or rich text: when a supported text file is selected you will be presented with a text field. Type in the separator that was used in the text to define sections. A common example might be a "#" character. Any line in the document that contains only the text entered into this box will be removed from the file and used to split the results in two. This process continues, further splitting the work into subsequent binder items, until all of these lines have been processed. A portion of the first line will be used to generate a binder title for each chunk.

2. FDX and FCF: Final Draft files can be imported using this tool, and when they have been selected the interface will change, giving you a selection of elements to choose from. You can select any one element to split by, such as scene breaks. The imported Final Draft file will be split into multiple binder items at these break points, and the script formatting will be converted to Scrivener's internal formatting. The sluglines will be used to name the individual scenes imported.

3. Fountain: Plain-text screenplays written using the Fountain markup syntax will be automatically split up by scene, when the file uses the ".fountain" extension and is chosen by this tool. The sluglines will be used to name the individual scenes imported, unless scene titles have been specified. Additionally, scene descriptions will be placed into the synopsis card, if discovered.

11.1.8 From Index Card for iPad

If you export projects from Index Card to Dropbox, you will get an .indexcard file. It is possible to drag these files directly into the Binder to import them. Each card will be imported as a separate document in Scrivener. If you would like to keep your Index Card projects synchronised with these items in Scrivener, then see Index Card for iOS (section 13.4).

This method does not have an associated menu command, it is purely a drag and drop method.

11.1.9 From Scapple Documents

Scapple[1], the freeform text editing software from Literature & Latte, gives you an easy to use interface for roughing out a new idea.

At some point, it may be advantageous to move your idea from Scapple into Scrivener. There are three ways of doing so:

1. If individual notes are dragged from an open Scapple document into the Scrivener binder, or into a freeform corkboard, they will be converted into individual Binder

[1] http://www.literatureandlatte.com/scapple.php

items, one per note. When dropping into a freeform corkboard, they will be placed spatially within the freeform corkboard in accordance with their relative positions in the Scapple document (this cannot be precise, since index cards are static in size and Scapple notes change their height depending on how much content they have within them). If dropped into a standard corkboard, or directly into the Binder, the spatial properties will not be maintained. This will be useful in cases where the conversion from notes to index cards is not advantageous and results in a lot of overlapping.

2. When bringing in a Scapple document that is not intended to be converted into individual items, one per note, you would want to export as a text document from Scapple (using the **File ► Export ►** menu in Scapple), and then import that as you would import any other text document into Scrivener. If you find this results in a confusing order, it might be better to use the first method, and then once you have the order set up correctly in the binder, select all of the imported items and use the **Documents ► Merge** menu command to concatenate them into a single document in the Binder.

3. If a Scapple document is dragged into the Binder (excluding the Draft), it will imported as a file, rather than as notes. On Mac OS X 10.7+, a Quick Look preview of the document will be displayed for you in the editor when you view it. Otherwise the typical file icon will be displayed. You can now open it as you would any other unsupported media format via the **Documents ► Open ► in External Editor** menu command, or clicking the application icon in the footer bar of the editor. This will be the most useful choice if the contents of your planning session are not meant to form a literal foundation for the work you will be doing in Scrivener, but rather as a reference to build from.

11.2 Scrivener Services

Scrivener installs several services that aid in grabbing text from other applications and getting them straight into Scrivener without having to worry about manually opening Scrivener and then pasting in the text yourself. The Services menu is not found in the main menu bar, but in the Services submenu of the application menu (which will be named according to whichever application currently has the focus). In all cases, you will need text to be selected for the appropriate services to become available. In versions of OS X prior to Snow Leopard, the menu options will appear greyed out, but in all newer versions, they simply won't appear until text is selected.

> **Services never show up**: In some cases, you will need to log out of your account and back in, after installing Scrivener for the system to properly register the services.

Where the clipped text shows up will depend in part on the service chosen. In all cases, the active project (the last project in use, even if Scrivener is in the background) will be used as a target, and in some cases the active document or split will be used as the target. In all cases, you must have at least one project open for services to work. If you wish to collect text into Scrivener, but do not yet have a project created, you can use the Scratch Pad Panel (section 11.3) instead.

Each method has an alternate form that will bring the selected text in unformatted. This can be useful when clipping text from the web, which often has inappropriate text colour and other formatting applied to it. All methods include an optional titling prompt. If you supply a separator or title, this will be placed into the document separating it from whatever content already existed. When using the formatted services, this title will be emboldened.

The following methods are available:

Append to Current Notes Targets active document split.

Selected text will be appended to the active document's Notes field, which may not appear to do anything unless you have the Inspector open.

Append to Current Text Targets active document split.

Selected text will be appended to the active document's main text area. If the current document in Scrivener is not one that can hold text (for instance, if it is an image document), the Scrivener icon will bounce and Scrivener will display a warning panel telling you that you cannot append text to any open documents, and asking if you would like to create a new clipping for the text instead.

Make New Clipping Targets active project.

Creates a new text document in the active project in Scrivener from the selected text in the active application. Note that all new clippings are placed inside a "Clippings" folder which will appear right at the bottom of the binder. This service is slightly different from the above two in that the title you provide will be used to name the clipping document that is created in the binder. A default, date-based title will be provided in case you do not wish to bother with naming them individually.

It is possible to adjust whether or not you are prompted for a title, or whether Scrivener comes to the foreground after using a Service, in the General preferences tab (section B.2).

11.3 Scratch Pad Panel

The scratch pad is a universal Scrivener tool that is not tied to any particular project. You can use it to jot down notes that don't fit anywhere else, or that arise whilst you are looking at another application. You can access it from the **Window ▸ Show (Hide) Scratch Pad** menu (⇧⌘–**Enter**), or you can right-click on the Scrivener icon in the Dock and select "Scratch Pad" (which will bring up the scratch pad over the currently active application). The ⇧⌘–**Return** shortcut is universal to your Mac, meaning you can use it to toggle the window while working in other programs, so long as Scrivener is running in the background.

The scratchpad is a floating panel, which means that you can have it open over all windows in Scrivener, and even while you work on other documents or even other applications.

The top pane contains a list of all the note documents you currently have stored in the scratch pad (by default there will only be one untitled note). New notes can be added by clicking the **+** button in the lower-left, and notes you no longer need can be deleted with the delete button next to it. Deleted notes will be moved to the system trash, so if you make a mistake, you'll find your notes by clicking on the trash can in the Dock.

When projects are open, the **Send to Project...** sub-menu will provide you with a list of all opened projects. There are two ways of bringing notes into the project:

1. *Append Text To*: The contents of the selected note will be appended after any existing text of the document you select in the project sub-menu. A list of your entire binder will be arranged so you can easily select any text item. Note that media files cannot be used since they cannot have text appended to them.

2. *Import as Subdocument of*: A new document will be created beneath the selected document. This sub-menu operates in a similar fashion to the above, though it will allow you to select any of the items in the binder since all types can contain children. The name of the scratch pad note will be used to populate the title field for the new document.

Since the scratch pad does not store your notes in any projects, it uses ordinary RTFD files, which by default will be stored in your Documents folder under "Scrivener Scratch Pad Notes". These are regular files that you can edit with other programs. You can even create your own rich text (or plain text) files in this directory, making it possible to add notes while Scrivener is closed.

To change where notes are stored, and the system-wide shortcut, use the General preferences tab (subsection B.2.5). Pointing your scratch pad folder at a synchronised or

shared location (like a DropBox or Google Drive folder) means you can share scratch pads between Scrivener installations if you use multiple computers.

The Windows version of Scrivener uses RTF as its format, rather than RTFD. If you wish to share scratch pads between platforms, you will want to change the default setting in the General preferences tab (subsection B.2.5). Existing scratch pad documents that originated from the Mac will need to be converted by hand using a tool like TextEdit.

11.4 Print as PDF to Scrivener

If the information you wish to import into Scrivener is locked in a format that cannot be used, a somewhat common way of capturing this information is to print the document from the source application, and when the print dialogue appears, use the PDF drop-down menu to select the target application. When you first ran Scrivener, it installed the necessary mechanism for this to work. You should see an option in that list to "Save PDF to Scrivener". The source application will assemble the print, save it as a PDF file, and then transfer that file to your active project. The imported PDF file will appear in your Research folder.

TIPBOX: **Mac App Store Installation**: If you purchased Scrivener through the Mac App Store, you will need to install this by hand, as the App Store guidelines do not allow our software to handle this for you. To install the PDF service you will need to create an alias of Scrivener in your Applications folder, and then place that alias in your personal Library folder, under "PDF Services". If you do not see a folder by that name, you can create one from scratch. The Library folder will most likely be hidden. To access it, use the Finder's "Go" menu with the Option key held down, to see it. Whatever you call the alias in this folder is what you will see in the PDF print menu. So feel free to name it to something more appropriate like the default, "Save PDF to Scrivener".

Chapter 12

Project Planning

12.1 The Corkboard

The corkboard provides a familiar, visual way of viewing documents in your binder. You can arrange index cards in direct correlation with their order in the outline structure, or alternatively as a freeform corkboard where cards can be freely moved about without directly impacting the structure of the book. We will explore the various features in depth, shortly, but first let's take a look at the index card itself.

In Scrivener, every document you create is a document *and* an index card *and* a corkboard *and* an outline. This can be a little confusing at first, for in the real world, an index card clearly cannot also be a corkboard. In Scrivener, though, you can choose to view any document as a corkboard. Each index card on a corkboard represents a document held *within* the selected document in the binder that is represented by the corkboard itself. Another way of looking at it: the corkboard displays the subdocuments of the selected document as editable index cards. At the same time, the document displayed as a corkboard or outline is also an index card (which holds the synopsis of that document), and itself could also be viewed as an index card on a corkboard or a row in the outliner.

If that is as clear as mud, then read on.

12.1.1 So What are Index Cards, Anyway?

One of the principles that takes some adjustment to get used to is the relationship between index cards and the text of the book itself. You can type titles and text onto the card just like you would the real thing, but there is sometimes confusion as to why that text doesn't get dropped into the book.

So let's take a look at what an index card really is. There are seven different components to an index card, but depending on your preferences you might not see them all at once.

1. The document icon
2. The document location numeral (optional)

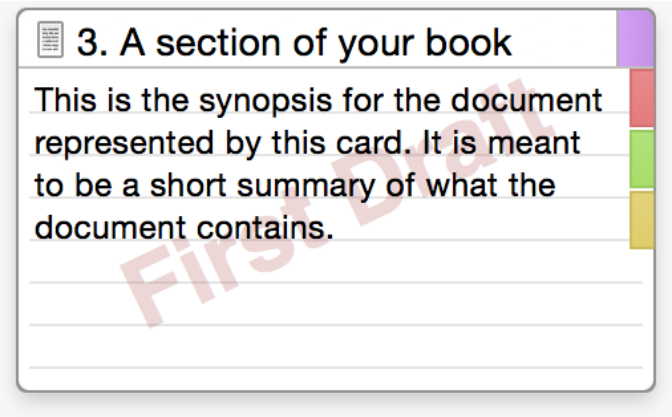
3. A section of your book
This is the synopsis for the document represented by this card. It is meant to be a short summary of what the document contains.
First Draft

Figure 12.1: A fully loaded index card using the rounded card theme.

3. The title of the document
4. The synopsis of the document
5. The colour label (colour swatch in top-right corner; optional)
6. The status stamp (diagonal text stamped across the middle; optional)
7. Keywords indicators (swatches along the right; optional)

Also, not shown in the screenshot (Figure 12.1), it is possible to tint the colour of a card based on its label. This can be used in addition to, or instead of, the colour chip in the upper-right corner. To tint cards by label type, use **View ► Use Label Color In ► Index Cards**.

At its most minimal, you'll only see the icon, title, and synopsis, the three core elements that cannot be removed. By default, keyword indicators will be shown, all other items must be enabled in the **View ► Corkboard ►** sub-menu. All of the options you select for corkboard appearance and label tinting will be saved into the project. If you prefer a certain look by default, consider creating your own custom starter template.

The icon, as you might have guessed, correlates directly with the icon that is visible in the binder. In most cases, this will be one of the variations of a folder or text icon, depending on its content status, but in some cases it might be a custom icon if you have supplied it with one, or if you are viewing a corkboard outside of the Draft, it might be some kind of media file icon.

The title is simply what you would expect: it is an editable name of the item as it appears in the binder, or in the header bar when you are editing it or view its corkboard.

The title is also often used by the compiler if it has been set up to consider it, but we'll come back to that in a later section.

The synopsis is meant to be a brief encapsulation of what the document's purpose is, but you can use it for whatever you like. Some people use it to keep track of the things they need yet to do, others keep highly visible notes about what they've written so far, and some don't even fill them in at all just leaving them blank or using Scrivener's automatic generation tools to form a synopsis from the first few lines of text. Whatever you end up using them for, it is important to realise that they are separate from the actual *text* of the document, and in most cases what you type into them will not appear in the final book. Use this to your advantage.

To edit either the title or the synopsis on a card, double-click in the text field you wish to edit. Once editing, you can use Tab and ⇧Tab to navigate between cards and their two editable fields, much like you would in a spreadsheet. To terminate editing, press the Enter key at any time.

12.1.2 Index Cards as Images

Instead of text, you can choose to display a graphic for the synopsis instead. This choice will be reflected in the inspector, as well as on the corkboard. There are two ways to set an item's synopsis to image mode. The easiest, if you have the graphic file already handy, is to simply drag and drop it into the index card in the inspector. This will automatically set the mode to image synopsis and import the graphic into the card for you. You may drag graphics from the binder, a text document in Scrivener, from the Finder or other software that allows one to export drag and drop graphics out of it. Alternatively, click the index card icon in the header area of the synopsis card in the inspector, or press ⌘–7 on your keyboard to toggle back and forth. A blank placeholder will be shown until you drag and drop an image into this spot.

Something to consider is that even if a card has been set to image mode, any text that was typed into it before will be preserved. You may not be able to see it directly, but it will be retained for tooltips, project searches and when compiling when the "Synopsis" checkbox is active in the Formatting compile option pane (subsection 24.11.1). Likewise, if you switch the card back to text mode, the image will be preserved in the background in case you ever switch back.

To remove an assigned image, click the X icon in the far right top of the index card in the inspector.

Image Size Options

Once placed, it is possible to adjust the display of the image to better fit the shape of the index card aspect ratio. Right-click on the image in the Inspector to access this menu.

You can only select one from each of the two sets of options. The first set lets you choose how the image will be scaled to fit:

- *Scale to Fit*: this is the default mode and it will display the image so that both the height and the width of it can be completely seen, even if this means adding extra padding to the sides.
- *Fit Horizontally*: scales the image so that the width of it will fit the frame, cropping pixels off on the top and bottom if necessary.
- *Fit Vertically*: likewise, but scaling the image so that the height of it fits the frame and the sides are chopped as necessary.

The second set lets you choose whether the image should be pinned to one side of the index card or another, and will the options will change to reflect logical choices depending upon your selection. For example, if horizontal fitting is chosen, the width of the index card will always be filled, so choosing to pin left or right would make no difference. In that case you choose between top and bottom. The default for both is to centre the image. This choice is irrelevant when "Scale to Fit" is selected, above.

12.1.3 Linear Corkboard

The standard, or linear, corkboard displays a linked representation of one level and section in your binder hierarchy. You can visualise this as a bit like looking at specimens on a slide. Each slide has a slice of a tree branch on it. To look at a different (whether deeper or higher) portion of the branch, you'll need to load a different slide. The corkboard displays *one* layer at a time, and by clicking up and down in the hierarchy you can view different layers.

In the linear corkboard, moving items around will change the actual order of those items in the binder. This makes it very useful for getting a "bird's eye" view of a section of your book, and enables you to affect changes upon the ordering of that book with simple drag and drop (you can also use the ^⌘–**Arrow** keys to move cards around in the direction corresponding to each arrow key). If you enable "drop-ons" in the Corkboard preferences tab (section B.4), you'll also be able to stack cards with other cards, just as you would drop items onto a folder. Many other appearance and behaviour settings can be adjusted in that tab.

12.1.4 Freeform Corkboard

While freeform corkboards have the same "single layer" method of looking at your binder structure, they do not have a rigid linked relationship with it. You can move

cards around freely and without any sort of grid constraints. To switch to freeform view, click the small "stack of cards" icon in the segmented control in the lower-right hand corner of the footer bar. Click the grid icon to its left to return to standard linear mode.

You can freely switch between modes at will. Scrivener will store the position of your cards in freeform mode even if you switch back to linear for a while. The two modes remain discrete from one another, though they do share some preferences which you can read about in the following pages.

Commit Order

Freeform mode can be useful for playing with an ordering idea without actually impacting binder order. You might wish to see how a sequence of scenes looks without actually changing the order and confusing things up in the binder. If you reach a point where you feel you have an improvement, you can choose to commit the freeform order back to the binder. Click the **Commit Order** button in the footer bar.

You will be given a few options to define your ordering style. Some work left to right, other right to left; this panel will let you apply the ordering no matter which way you work. Once you click the Okay button, nothing may appear to happen unless you were paying attention to the binder. Committing the order will never disrupt the cards position.

When using freeform corkboards for this purpose, you might find it useful to enable **View ► Corkboard ► Show Card Numbers**.

12.1.5 Corkboard Options

The Corkboard Options pop-up contains settings which are specific to each project. Changes made to this pane will only impact the active project they have been made in, letting you have two entirely different visual appearances for each major work. Any changes made will be saved with the project, meaning you can save your preferred defaults into a custom project template for future use.

To access the pop-up, click on the "Corkboard Options" footer bar button, as shown (Figure 12.2). You can click anywhere on the corkboard to dismiss this pop-up pane.

Since the interaction between some of these settings are interdependent, here is a useful guide for adjusting the settings based on what you want:

- A large corkboard that ignores the width of the editor: disable **Size to fit editor**, set the **Card across** to the desired corkboard width, and adjust the **Size** slider as desired.
- Cards that have a fixed size that wrap like words depending on the editor width: disable **Size to fit editor**, set **Cards across** to Auto, and adjust the **Size** as desired.

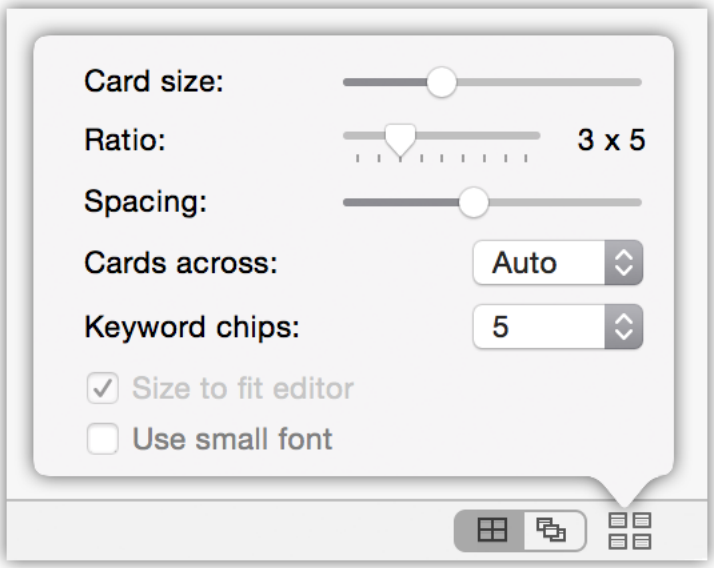

Figure 12.2: Corkboard options pane

- Cards that adjust their size according to the editor width: set **Cards across** to the desired amount and enable **Size to fit editor**. This is how the screenshot (Figure 12.2) has been configured. It will always print two cards per row, sizing them to be as large or small as necessary, depending upon the width of the corkboard in the editor.

Size There are two ways of arranging index cards within a corkboard. The first is to set the size of cards and then let the corkboard wrap the cards as they fit, the second is to provide a number of cards you always want to see in each row, and let the corkboard resize the cards to fit that number. When the latter method is in use (see below, for setting that), the Size control will be disabled.

Ratio Determines the size ratio between height and width. By default this will be 3 x 5, in order to emulate the appearance of real index cards. If you write very long or very short synopses however, you might find that adjusting this to produce shorter or taller cards will be of benefit.

Spacing The amount of space that will be drawn between index cards, both vertically and horizontally. To pack more cards into the display at once, move the slider toward the left. To spread out the cards and make them more distinct, move the slider to the right. *This option is only relevant to linear corkboards.*

Cards Across Set this to the number of cards you would like to appear in each row. Use "Auto" to have Scrivener determine this amount based on the size of the cards. When "Size to fit editor" (below) is disabled, it is possible for corkboards to grow larger than the editor size, which may be desired for some purposes, and will require horizontal scrolling in order to see the entire corkboard contents. You can choose any number between 1 and 10 cards across, or use the "Other..." option to specify any arbitrary amount. *This option is only relevant to linear corkboards.*

Keyword chips Set the maximum number of keyword colours to be "taped" to the right side of the index card. When a document has more than that amount assigned to it, all keywords below the specified point will be ignored. You may wish to adjust the *Ratio* to increase the height of the index card, if you want to view large numbers of keyword chips at once.

Size to fit editor This option is not available when "Cards Across" is set to Auto. When the Cards Across option is set to a number, this option will resize the cards to fit the current editor width. With this option off, the card size option will be used, and cards will be forced to wrap at the specified number regardless of the window size. *This option is only relevant to linear corkboards.*

Use small font The small font settings can be configured in the Corkboard preferences tab (section B.4). Note that since you have full control over this and the standard index card fonts, "small fonts" might not actually be any smaller than the regular font size.

> **Further Options**: There are a number of other appearance related preferences that you can set, which can dramatically alter the look and feel of the corkboard. Most of these options are included in the Corkboard preference tab (section B.4), but a few colour related options are also in the Appearance tab (section B.3).

12.1.6 Stacked Corkboards

When more than one container of any type has been selected, the corkboard will switch to a special stacking mode, displaying the contents of each container, separated by a line. By default, this will wrap each container's section as though it were a normal corkboard; cards will be displayed according to your view settings. You can also select between vertical or horizontal stacking by clicking on one of the left two buttons in the segmented control, found on the right side of the footer bar, near the view options button.

Horizontal mode will place all of the cards within each container on a single row, so you can easily view them sequentially. Vertical mode will do likewise, but reserving

one column for each stacked container. There is no functional difference between these modes save for aesthetics and how they use screen space. Both are useful for tracking the precise flow of index cards between a large selection of containers. You might, for instance, use it to compare how scenes in a novel are balanced between chapters, but using the label or keyword colours to scan for POV and character appearance frequency, or plot threads.

An optional numbering scheme can be applied with **View ► Corkboard ► Number Per Section** combined with the above option to "Show Card Numbers", this will start counting from one for each container stack. When disabled, cards will be numbered sequentially even across container sections.

12.1.7 Placing Images on the Corkboard

When working in an area of the binder outside of the draft folder, it is possible to import graphics and other media into your binder. They will be supplied with special icons, indicating the type of file they are. Graphics will be represented with a small iconic image of a sunset photograph.

When viewing a corkboard that contains graphics, they will be drawn on the corkboard as a photograph, rather than the typical index card. This behaviour can be changed in the Corkboard preferences tab (section B.4), if you prefer to see your summary synopsis for the image instead.

You may also change how any item at all in the corkboard is represented in a way which will always override the default. Even folders and standard text files can be represented by a photograph, or *vice versa*, photographs can be represented as index cards with a text synopsis. For further instructions on how to do this, read the Synopsis Card (section 19.1) overview in the chapter on the inspector.

This will only change the item's representation on a corkboard. The text synopsis will not be deleted, and can be edited using the outliner. The binder icon will not change. If your graphic files have text icons, this means they have been dragged into a text file, rather than imported directly. You can rectify this problem by dragging the image from the text file back into the binder.

12.2 The Outliner

Outliner mode shows all of the descendants of the current document along with select associated meta-data in a tabular format like a spreadsheet. The default configuration will show the title and synopsis in the main column on the left. You can edit by double-clicking into the text field you wish to edit. Once editing, you can move between titles

and synopses with the arrow keys, much like in an ordinary text editor or outliner. Press Enter to exit editing mode.

You can expand or collapse the outline by clicking on the small disclosure arrow to the left of the title. When viewing an outliner with only text documents, you may not see any arrows. All items can be expanded completely with View ▶ Outline ▶ Expand All. Conversely, all items can be completely closed with View ▶ Outline ▶ Collapse All. Additionally, you can hold down the Option key and click on any arrow to completely expand or collapse that portion of the tree.

Most columns that allow you to edit their data will provide controls for doing so.

- Checkboxes: an example, "Page Break Before" will present a checkbox that you can click to add a page break before that document when exporting. You can impact many checkboxes at once by holding down the Option key and clicking on a checkbox. All *visible* checkboxes will be impacted. This means if items have been hidden with their disclosure arrows in the outliner, they will not be impacted. However if you select a number of items first and Option-click on a checkbox, only those selected items will be modified.

- Labels and Status and Target Type will provide drop-down menus that you can use to adjust the meta-data for a row. For labels and status, If you wish to impact more than one row at once, select these rows first and then use right-click to access the contextual menu, instead of clicking directly in the outliner.

- Columns with editable text, such as the Title, custom-meta data, or Target column will not provide a control, but you can simply double-click on the field to change their value.

12.2.1 Managing Columns

To add or remove columns from the display, click the small "►►" button in the far-right column header bar, right-click in any of the column header names, or use the View ▶ Outliner Columns ▶ sub-menu. Column settings are saved per editor split. So you can set up an outliner to perform a particular function on the left side of your screen, and display extended information on the right side, just to provide an example. The column selection pop-over will remain focussed until you click elsewhere or press the Escape key, making it easy to manage multiple columns. If you would prefer to use the single-access menu instead, right-click on the "►►" button.

To change the order in which the columns are displayed, drag and drop the column header to the desired location. You can resize the column width by moving your mouse

between column header titles until the cursor changes to a double-pointed arrow, then click and drag to increase or decrease the width of the column.

Title Column: these three options specify how the compound Title column should appear. This column will be named "Title and Synopsis" when the "with Synopsis" column is checked.

- Title
  - and Synopsis
  - with Icons

Corresponding to the general meta-data: the first two columns allow you to edit the Label and Status of each row individually. The remaining three columns are read-only. Keywords will be underlined using the colour that has been assigned to that keyword.

- Label
- Status
- Keywords
- Created Date
- Modified Date

Statistics: the first two options show the word or character counts for each item in the outliner. The second set will not only show the statistics for that row, but will sum up all of the items descending from that row as well. A folder itself might have no words in it, but if it contains five text files with 1,000 words each, the folder's row will display 5,000 in the Total Word Count column.

The word count columns will colour the text according to your progress bar colours. So if you prefer to not turn on the progress bars (discussed below), you can still get a rough idea of how far you are form your goal, based on the colour of the text.

- Word Count
- Character Count
- Total Word Count
- Total Character Count

Further options from general meta-data: the checkboxes displayed in these columns are directly linked to their respective options in the Inspector, general meta-data section.

- Include in Compile
- Page Break Before
- Compile As-Is

Targets and progress tracking: as with statistics, these show target tracking information for each item in the outliner, and the "total" variants will add up the combined statistics for all descendants as well. If a folder has five text documents, each with a goal of 1,000 words, the Total Target column will display 5,000. If the sum total of all words in the descending documents is 2,500, the Total Progress tracking bar for the folder will be filled to 50%.

- Target
- Target Type
- Progress
- Total Target
- Total Progress

Any custom meta-data fields you have created will be listed at the bottom of the menu as well.

12.2.2 Special Columns

In addition to the standard columns, there are two choices which impact the main Title column. They are indented beneath the Title option in the column menu.

- *and Synopsis*: embeds the text of the index card synopsis field beneath the title. With this option enabled, the title will be emboldened and you can edit both Title and Synopsis together right in the outliner. The **Show (Hide) Synopsis** button in the footer bar provides a shortcut to turning this special column on or off.
- *with Icons*: When disabled, item icons will be removed from the outliner display, producing a cleaner, more "text like" appearance.

The Title (and Synopsis) column is also special not only in that it provides compound information, but if it is the only column present, it will automatically fill the entire width of the outliner.

12.2.3 Sorting by Columns

To sort by columns, click on the column header. The first click will sort ascending, clicking again will sort descending, and a third click will turn off sorting.

This form of sorting is a way of viewing information in the Outliner. It will not impact the underlying order of the items in the Binder, so you can safely use it in conjunction with other methods of gathering items or viewing them, such as search results or multiple selections, without fear of disrupting your book structure.

If you do indeed wish to change the underlying Binder order to match the outlining sort order, you can do so by selecting all of the outliner rows you wish to change and dragging them back into the folder they reside in, in the Binder or back into the folder they are indented beneath in the Outliner itself. When items are dragged (even back to their same location), their order is remembered, so as long as the sorted items come from the same folder, dragging them back to that folder will re-order them in according with their visible order in the Outliner (or Corkboard, for that matter) view. This cannot be used to re-order many items within sub-folders at once. You'll need to re-order each folder individually.

In the case of the Title (and Synopsis) column, sorting will be performed by item title, not the synopsis content.

12.3 Project Notes

As your project begins to take shape, you will no doubt have many ideas that you wish to capture, but nowhere concrete to put them yet. Documents all have a separate note pane which can be accessed in the Inspector (section 19.4), but if you don't even have a stable outline built yet, there are no documents to put notes within. There are a number of ways you could approach this problem, such as keeping notes as documents in the Research folder or somewhere else outside of the Draft or even in the Draft with their respective "Include in Compile" flags disabled.

You might also find the Project Notes tool to be useful, as it is accessible from everywhere in the program. Project notes, like document notes, are a rich text field. That means you can format within them however you please, even drag pictures into them. Project notes can have different styling than document notes, making it easy to tell which you are currently viewing or editing. These can be set up in the Appearance preferences tab (section B.3).

Their primary advantage is that they can be viewed from the Inspector, no matter what you are viewing, and even make an appearance when what you are viewing what would ordinarily have no Inspector data, like the Draft or Research folders.

> **Project Notes and Exporting**: Project notes are not set up to be easily exported. There is no way to compile them at all, without copying and pasting their contents into a Binder document. Document notes can be included in various compile options, but project notes cannot; nor can they be exported via any menu commands. So use project notes only for things which are strictly internal to the project. If you intend to export or compile notes at some point, it might be easier to take the original suggestion of using documents in the binder.

To demonstrate, open the inspector if necessary (click the blue 'i' button in the toolbar, or press ⌥⌘-I) and click on the Draft folder. The inspector will be replaced by a single pane titled "Project Notes" (if that didn't happen, you might be viewing one of the index cards in the corkboard that pops up. Just click once in the background of the corkboard to de-select any cards.) Type in a sample line into Project Notes so we can track it around the interface as we explore this feature.

This field is not only accessible from the Draft, but any document can view the project notes in their Inspector as well. Try clicking on one of those index cards in the Draft corkboard and make sure the left-most button in the top of the inspector is selected (the one that looks like a little notepad). By default, this will show you the document notes for that selected card. You can click on the header bar where it says "Document Notes" and select "Project Notes" instead. The sample line that you typed in above will become visible and the background colour will change. You can easily flip between notes with ⌘-6 while the inspector is open.

Project notes can also be viewed in a floating window. To view the project notepad, invoke the **Project ▶ Project Notes** menu command, or press ⌥⌘-P. The text area is a mirror copy of what you see in the Inspector.

See also: The Inspector (subsection 5.1.5).

12.3.1 Managing Project Notes

Open project notes by using the **Project ▶ Project Notes...** menu command or by pressing ⌥⌘-P.

The sidebar along the left will display the available notepads in this project. Along the bottom you'll find a **+** button. Click that to create a new tab. Now you have two project notebooks. These are completely independent from one another, and so can be used to store different types of notes.

To rename a notepad, double click on its name in the sidebar. Notepads can also be re-ordered by clicking and dragging on them.

Delete the selected notebook by clicking the **-** button.

12.3.2 Using Multiple Notepads in Inspector

With multiple notepads you'll probably want access to them in the Inspector as well. To see how this is done, view a document and open the Inspector, click on the notes header again and you'll see the new notepads you created.

As always, use ⌘–6 to flip between them all, and now ^⌘–6 is also available to rotate backward through the list. Both of these commands wrap around the list so if you get to the top or bottom they will keep going in the same direction in an endless cycle.

12.4 Setting Default Project Formatting

Sometimes a project will need its own visual font and ruler settings. Ordinarily, all new files you make in all projects will conform to the settings in the Formatting preferences tab (section B.8). Using **Projects ▸ Text Preferences...**, you can supply an override to a specific project, so that its new documents take on a custom appearance.

> **Scriptwriting and Fonts**: when using Scrivener to compose scripts, the font for the script will be selected for you based on the script settings, not the application settings or this override. See Scriptwriting (chapter 20) for further details.

First, enable "Use different default formatting for new documents created in this project", with the checkbox at the top, and then proceed to use the mock editor to set up your preferred styling for this project. This works just like the Formatting pane, or if you haven't used that before, much like the standard text editor. The **Use Current** button provides a handy way of importing settings from the current text editor at the cursor position. So if you've already set up a document to look the way you want it to, using this button can save some time.

Use different font for footnotes This also overrides the application preferences for an alternate footnote font. This setting will impact both inline and linked footnotes (chapter 18). It will work even if you are not setting a custom font for footnotes in the main application preferences.

Use footnote marker An alternate way of referencing linked footnotes, which places a custom marker after the selection or current word, rather than highlighting it. In documents that make use of heavy notation, this can help keep the editor looking clean. The marker used can be typed into the text field.

If you prefer this behaviour for all of your projects, clicking the **Make Default** button will force your setting on all projects, even those you've already created.

Read more about this feature in Linked Footnotes (subsection 18.2.3).

Chapter 13

Cloud Integration and Sharing

It is becoming increasingly important to have the ability to make your work instantly available everywhere you go, no matter what device you might currently be using. Sharing files amongst a circle of colleagues, proofers, and editors is also becoming common. In conjunction with the Simplenote[1] network, you can distribute copies of text items from your binder to your note network, and access them from anywhere using portable devices or other Simplenote-aware applications. In addition, a generalised tool using simple files and folders has been provided, which can be integrated with file sharing services such as Dropbox[2], which can in turn be accessed by numerous text editing applications on mobile platforms.

Synchronisation should always be done with care, no matter what method you use. Computers will do exactly what we tell them to do, even if the outcome is not what our original intention was. Whenever using automated syncing tools, be sure to double-check your settings, and backup frequently. Scrivener uses automatic Snapshots to protect your work by default. This can mean a lot of automated Snapshots may build up over time, but given the unpredictability of all the many factors involved, it's generally a good idea to leave it turned on.

13.1 Simplenote

The Simplenote Sync wizard allows you to easily sync documents in your Scrivener project with Simplenote, so that you can work on or create documents while on the go using an iPad, iPhone, iPod touch or the Simplenote web application.

To use this feature, you will need to create a Simplenote account by downloading the Simplenote app to your iPad, iPod or iPhone.

13.1.1 How It Works

Simplenote is a general purpose note-taking system and knows nothing of Scrivener or Scrivener projects. When uploading documents to Simplenote, Scrivener therefore places

[1] http://simplenote.com/
[2] http://www.dropbox.com

some information on the first line so that it is able to find and recognise these documents again later, when you want to sync them. The information Scrivener places on the first line is formatted like this:

```
MyDocumentName (MyProjectKeyword [ID])
--
```

"MyDocumentName" represents the document title as it appears in the binder, "MyProjectKeyword" is the keyword you assign to the project in this wizard (see below), and "ID" is an internal numerical ID that Scrivener assigns to all documents, which you won't normally see. In practice, it might look like this:

```
The Grand Finale (MyGreatNovel [756])
--
```

These three pieces of information in combination allow Scrivener to sync documents with Simplenote. You should therefore not edit this line in Simplenote. In particular, if you edit the project keyword or numerical ID, Scrivener may not recognise the Simplenote text as belonging to the current project, or worse, it may try to sync it with the wrong document in Scrivener. It is thus generally recommended that you leave this line alone. However, you can edit only the document title (the part before the parentheses), which will update the title in Scrivener when synced, if you really want to.

13.1.2 Getting Started

Login details The Simplenote Sync wizard will first prompt you to enter your login details. Enter the e-mail address and password which you used to register for Simplenote. You can check "Remember in keychain" to have your login details remembered for the future.

Project keyword Along with your Simplenote login details, you must also enter a project keyword if you have not already chosen one. This is used by Scrivener to identify documents in Simplenote that are associated with the current project. You should choose a keyword that is meaningful to the project, easy to remember, and that is unlikely to occur on the first line of documents in general. For instance, you might use a wordsJammedTogether version of your project title, such as "MyGreatNovel".

Do not use (parentheses) or [square brackets] as part of your keyword. Since these are used by Scrivener to identify the salient portions of the note's identification, adding additional brackets into the token can cause the sync to malfunction.

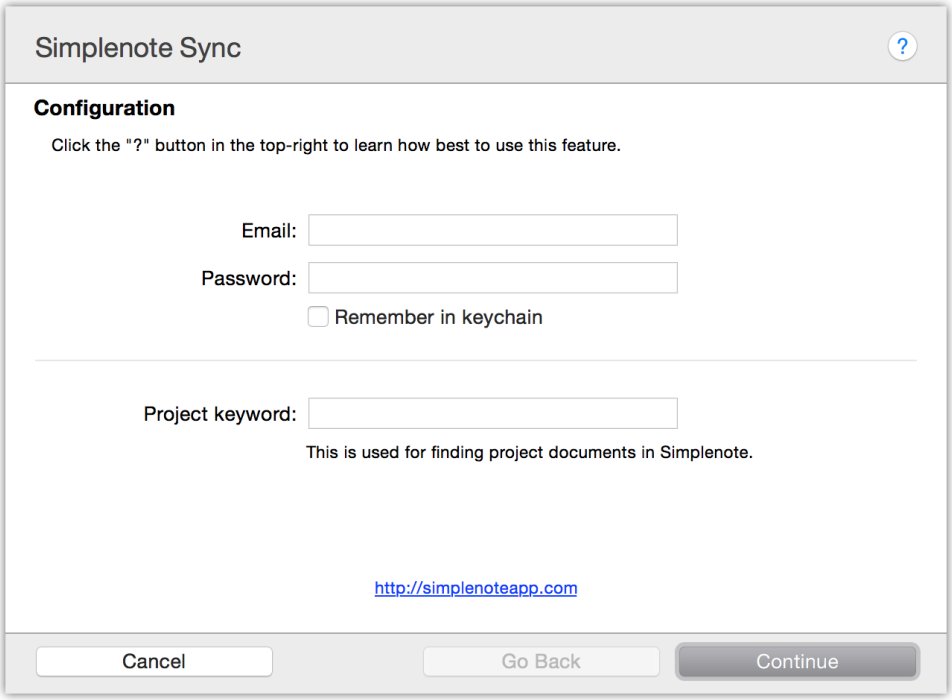

Figure 13.1: Simplenote Sync: Configuration

NOTE: It is important to understand that Scrivener has no way of knowing which documents in Simplenote are associated with which Scrivener project without this keyword, so it is up to you to choose a unique keyword for **each project**. If two projects have the same keyword, you could run into problems, with each trying to sync the other's documents in Simplenote.

The project keyword is also very useful for viewing notes in Simplenote, because you can use it as a search term to filter Simplenote so that only documents from your project are shown in its file list. In addition, Scrivener will add this project keyword as a tag to each document, so you can view by tags to isolate notes corresponding to a specific project.

Once you have entered your login details and ensured that a project keyword has been supplied, click on **Continue**. You won't be prompted for the project keyword or login details next time you sync the project with Simplenote, but if you need to change anything you can return to this pane by clicking on the **Change Settings** button in Step 1. Note that the project keyword is shown in the top right of the wizard in steps 1 and 2.

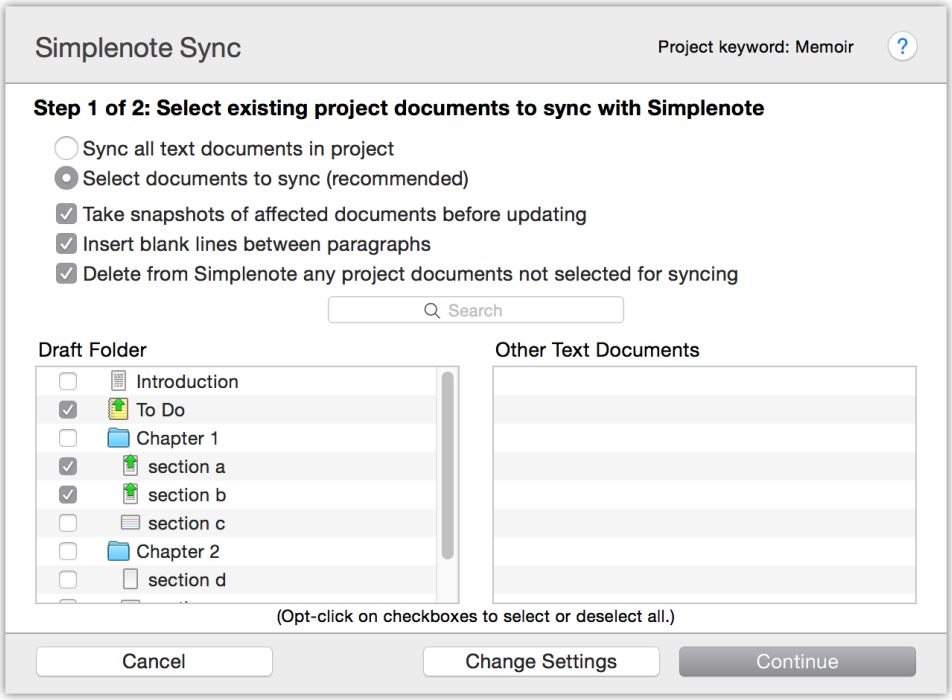

Figure 13.2: Simplenote Sync: Step One

13.1.3 Step 1: Synchronizing Existing Documents

You will next be asked to choose which documents in the project you wish to sync with Simplenote. You can choose to sync all documents in the current project or to specify individual documents. It is generally recommended that you don't sync all documents in a project, because this can clutter up Simplenote and severely slow down synchronisations. Because Simplenote notes are stored on the cloud (that is, online), Scrivener has to download them into memory every time you sync, so the more notes you have in Simplenote, the longer the sync process will take.

If you choose to identify the documents you wish to sync (as recommended), you are presented with separate lists for documents contained in the Draft folder and all other text documents in the project (Simplenote is text-only so you can only choose text documents). You should go through and tick the documents you wish to upload to Simplenote for editing while on the go.

All documents that have already been uploaded and exist in Simplenote will be ticked for syncing automatically. The icons (Table 13.1) next to the document titles will tell you what will happen when you choose to sync.

The following options are available.

Table 13.1: Simplenote Status Icons

Icon	Meaning
Green check	The document is up to date both in the project and on the Simplenote servers—the version in the project is the same as the version in Simplenote.
Up arrow	The document will be uploaded. This indicates either that the version of the document in the Scrivener project is newer than the one in Simplenote or that it has not yet been uploaded at all and is now selected for syncing.
Down arrow	The document will be downloaded. This indicates that there is a newer version of the document in Simplenote which will be downloaded in order to update the version in the project.
Green asterisk	The document needs updating in Simplenote. This green asterisk appears if you choose (by deselecting) not to sync a document that exists on the Simplenote servers but which has been edited in Scrivener since the last sync. If you select to sync the document, the asterisk will turn into a green "up" arrow to indicate that it will be uploaded.
Red asterisk	The document needs updating in Scrivener. This red asterisk appears if you choose (by deselecting) not to sync a document that has a newer version available on the Simplenote servers. If you select to sync the document, the asterisk will turn into a red "down" arrow to indicate that it will be downloaded.
No icon	The document is not selected for syncing and does not exist on the Simplenote servers.

Take snapshots of affected documents before updating When ticked, Scrivener will take a snapshot of documents being updated before overwriting them with the new version of the text. This ensures that you can return to the earlier version if something goes wrong with the sync process. It is therefore highly recommended that you keep this ticked, as it provides a good degree of protection against losing work in cases where you have inadvertently made changes to the documents both in Scrivener and in Simplenote between syncs.

Insert blank lines between paragraphs For most writing, you'll want to leave this on, as plain-text documents do not show any space between paragraphs unless they literally have a space. If your already includes those spaces, or has precise line based formatting, disabling this option will cause Scrivener to leave the line spacing alone in the document. You will likely also want to disable the opposing option in step two, below, which strips out extra lines.

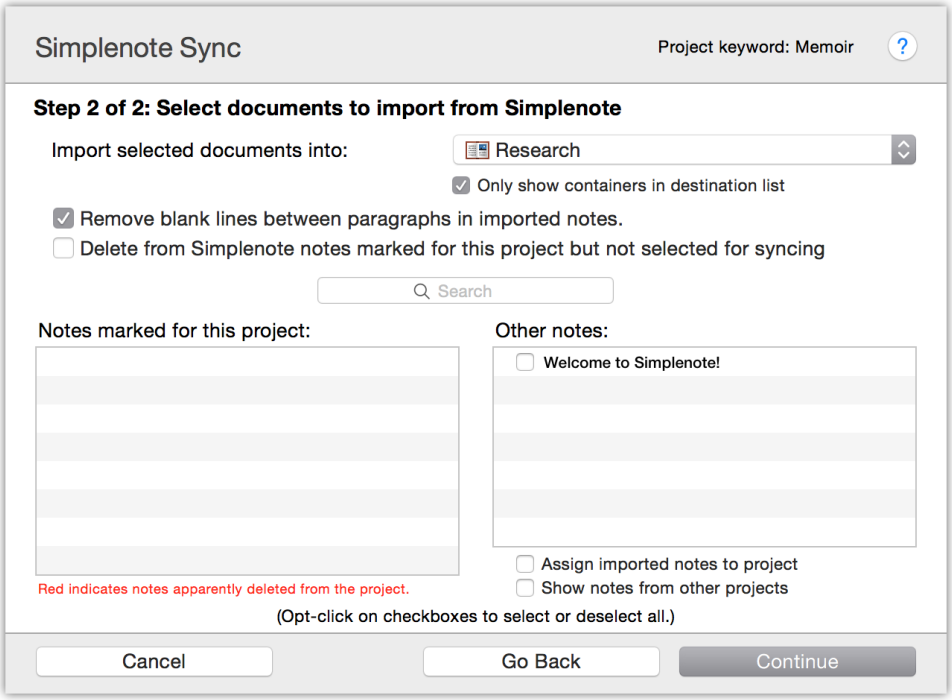
Simplenote Sync
Project keyword: Memoir
Step 2 of 2: Select documents to import from Simplenote
Import selected documents into:
Research
Only show containers in destination list
Remove blank lines between paragraphs in imported notes.
Delete from Simplenote notes marked for this project but not selected for syncing
Search
Notes marked for this project:
Other notes:
Welcome to Simplenote!
Assign imported notes to project
Show notes from other projects
Red indicates notes apparently deleted from the project.
(Opt-click on checkboxes to select or deselect all.)
Cancel
Go Back
Continue

Figure 13.3: Simplenote Sync: Step two

Delete from Simplenote any project documents not selected for syncing When ticked, Scrivener will delete from Simplenote any files that were previously uploaded from this project that are no longer ticked for syncing. This is recommended in order to avoid clutter and slowing down Simplenote.

Once you are happy with the documents selected for sync, click **Continue**.

13.1.4 Step 2: Importing Documents From Simplenote

The final part of the wizard allows you to choose to import documents that exist in Simplenote but which do not yet have corresponding files in the project. Given that Simplenote is a general purpose note-taking system, there may be many notes in your Simplenote account that you do not wish to import, including notes belonging to other Scrivener projects. This pane of the wizard therefore contains two lists.

The first list contains documents that seem to have been marked as belonging to this project. These are documents that contain the project keyword (see above) on the first line of the note enclosed in brackets, and which were not detected as already belonging to the project. This makes it very easy to create a document in Simplenote and mark it

for importing into a particular Scrivener project: just write the document title followed by the project keyword in brackets on the first line of the note, e.g:

```
MyNewNote (MyProjectKeyword)
--
```

(The two hyphens on the second line are optional.)

Note that there is no numerical ID in square brackets in this case because that is something only Scrivener can assign (and when you sync these documents, Scrivener will indeed add a numerical ID). Do not try to guess and assign your own. The documents in this first list are usually all ticked by default. The exception are documents that Scrivener thinks might have been deleted from the project but are still on the server—such documents appear in this list but are coloured red and are not ticked. These are documents that contain the project keyword on the first line and a document ID in square brackets, but for which the ID does match any existing documents in the project. If you see notes coloured red in this list, the most likely explanation is that you synced them in the past but have since deleted them from the project. If you leave them unticked and have "Delete from Simplenote notes marked for this project but not selected for syncing" checked, then these notes will be removed from the Simplenote servers too.

The second list contains all other Simplenote documents, and these are not ticked by default. You can use this list to assign notes to the project, if the "Assign imported notes to project" option is checked, below the list. If you do not assign them, they will be imported but the original note will not be set up to sync with Scrivener in the future.

Additionally, you can enable "Show notes from other projects" to see items which are already assigned to other Scrivener projects that are syncing with Simplenote.

The following options are available:

Import selected documents into At the top of the "Import" pane you can choose the folder in the project into which imported documents should be placed (the Research folder is used by default). **Only show containers in destination list** limits the documents shown in the list to containers (folders are always considered containers, even if empty); if unchecked, you can choose to import documents as the subdocuments of any file in the project. (If you are unsure, just leave it ticked.)

Remove blank lines between paragraphs in imported notes Allows you to choose whether or not empty lines should be removed in imported documents: because Simplenote is plain text, it is common to separate paragraphs using empty lines, but in Scrivener paragraphs are less likely to need blank lines between them. (Note that

when uploading documents to Simplenote, Scrivener automatically places blank line padding between paragraphs to make them easier to read in Simplenote.)

Delete from Simplenote notes marked for this project but not selected for syncing
When checked, any notes that appear in the "Notes marked for this project" list that are not ticked for syncing will be removed from the Simplenote servers. (Given that this could result in the deletion of the only copies of such notes, you will be asked to confirm this deletion during sync.)

Once you are happy with the documents you have selected for synchronisation and import, click on **Sync**. You can choose **Go Back** at any point to make changes.

13.1.5 Using Simplenote Tags

When syncing files, Scrivener will assign two tags to each file:

1. The tag: "Scrivener", which will make it easy to find all files associated with any Scrivener project.
2. A tag corresponding to the project keyword you selected above, making it easy to filter your note list in the Simplenote application by project.

You can also add new documents to the project from Simplenote by simply adding the project keyword tag to that document. When using this method, you needn't supply the project keyword in parentheses after the title; Scrivener will add it for you.

Another way of assigning notes to a project is to use the "Other notes" browser, in Step 2, to find the note, tick its checkbox, and then also the "Assign imported notes to project" checkbox, below this list.

13.1.6 Clearing Out Simplenote

When you've reached a point in your project where you no longer require external syncing, or would for any other reason like to clear off everything belonging to your project, it is very easy to remove everything from the server pertaining to the project.

Before following these instructions, it might be a good idea to run one last sync to make sure everything is up to date. Then, simply make sure the "Delete from Simplenote notes marked for this project but not selected for syncing" is enabled in Step 1 and Step 2, and then **⌥Click** on any file checkbox currently enabled. This will de-select everything and wipe out the related notes on Simplenote when you click the **Sync** button. Note that nothing in the project will be impacted when you do this, it will only clear out the associated notes on the Simplenote server.

13.1.7 Limitations

Formatting Note that Simplenote, like most mobile note-taking solutions, can only store plain text, so all formatting—bold, italics, line spacing and so forth—will be stripped in the versions of documents stored in Simplenote. Thus, when updating documents in a project with corresponding versions that have been edited in Simplenote, the following formatting limitations apply.

- Scrivener maintains the formatting of paragraphs within a note that have not been changed.
- Any paragraphs that have been edited in Simplenote will lose their formatting in Scrivener.
- Images will be stripped from documents that have been edited in Simplenote.
- To avoid having inline notation become confused with standard text, Scrivener will export inline annotations by wrapping them in double-parentheses ((and)). Inline footnotes will be similarly wrapped in curly braces {{ and }}. These work in both directions. If you type them into a paragraph using your text editor, they will be converted to their respective type of notation upon syncing.
- Inspector comments and footnotes will not be exported. If the retention of this information is important to you, it is recommended you use inline notation with plain-text. You can easily convert your notes to inline with the Format ▸ Convert tools.

Number of Notes At the time of writing, Simplenote is limited to syncing 2,000 notes a day. Although this sounds a lot, if you end up syncing the entire contents of ten projects that contain 200 text documents, you can soon end up with a lot of notes in your Simplenote account. It is therefore strongly recommended that you only sync the documents you need for working on while away from Scrivener, and that you ensure the option is ticked to trim documents that are not being synced. This will also help ensure that syncing remains as fast as possible.

13.2 Synchronised Folders

Scrivener lets you prepare a special external folder on your disk, which it will use to keep the text contents of a project up to date if (but not when) changes are made to the contents of this folder. Because it works with simple flat list of files and a few built-in folders, the format is uniquely suited for integrating with various folder sharing services such as Dropbox, SpiderOak, iCloud, and others. Using this feature, you can share your work in

progress with collaborators, agents, or editors, and later read back any changes they have made directly into your project. Additionally, you could use the feature in conjunction with mobile applications and devices, making it possible to edit your project's content anywhere, and later automatically merge those edits with your main project.

> **Important Note**: This feature is *not* intended for keeping two versions of the same project in sync and should not be used in such a manner—you should never try to sync two different projects with the same folder. Rather, this feature is intended to allow you to edit or share project files with other applications or on other platforms, and then to have any changes made to the shared files reflected in your Scrivener project. It also cannot be used to perform structural or outline order modifications, just simple text content modifications.

13.2.1 Setting it Up

To access the feature, select the menu item, **File ► Sync ► with External Folder...**. A sheet will drop down in the active project with some basic settings.

The first thing you will need to do is select a new, empty folder which Scrivener will take over and use for syncing files with this project from this point forward. Click the **Choose...** button, and navigate to the location where your cloud service keeps your computer synchronised. For Dropbox users, this will be the Dropbox folder in your home directory. Keep in mind that the folder you create will need to be dedicated to this Scrivener project, and so must be initially empty. When you sync documents with this folder, Scrivener will create one or more of the following three subdirectories to store the synced files:

- "Draft" - stores documents contained inside Scrivener's Draft folder.
- "Notes" - stores text documents contained elsewhere in the project.
- "Trashed Files" - stores documents that had been synced but have since been removed from Scrivener, or which had some sort of conflict. You should occasionally review these files and trash them if they are no longer required.

It is not possible select a folder which has already been used to synchronise another Scrivener project! In this case, another project can very well mean the same exact project you just copied to another computer. This will most often be an issue when you use the project on more than one computer, and your account name on those computers are different (thus making the full path different). When working in situations such as these, it is best to turn off automatic sync on project open and close, and run it manually from your primary computer. There is never a reason to use this feature to keep two

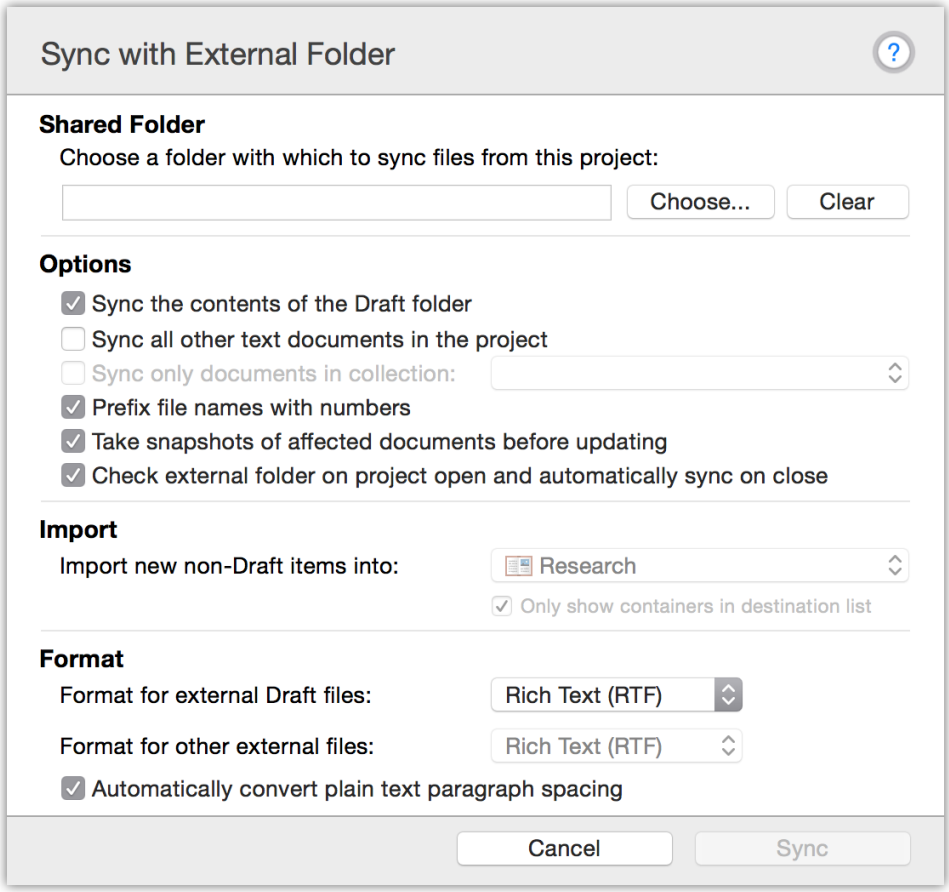
Sync with External Folder
Shared Folder
Choose a folder with which to sync files from this project:
Choose...
Clear
Options
Sync the contents of the Draft folder
Sync all other text documents in the project
Sync only documents in collection:
Prefix file names with numbers
Take snapshots of affected documents before updating
Check external folder on project open and automatically sync on close
Import
Import new non-Draft items into:
Research
Only show containers in destination list
Format
Format for external Draft files:
Rich Text (RTF)
Format for other external files:
Rich Text (RTF)
Automatically convert plain text paragraph spacing
Cancel
Sync

Figure 13.4: Sync External Folder Sheet

copies of the same project up to date. If you need to work on your project from more than one machine, it is best to copy the project completely. The reason for prohibiting this is that even when working from the same exact project, it is all too easy to cause problems between copies of the project, which will inevitably differ over time, internally.

Using this feature in an attempt to merge projects, or to have more than one person using a copy of the project, could result in lost data or confusing results.

> **Saving changes without syncing**: If you merely want to adjust your sync settings (commonly to enable or disable automatic checking on project open and close), you can hold down the Option key in the set-up window to change the **Sync** button to a **Save** button. Clicking this will update your project settings without running sync.

This feature provides the following options:

Sync the contents of the Draft folder Enabled by default, the entire Draft folder will be kept in sync with the folder when this is on.

Sync all other text documents in the project Keep the rest of the project up-to-date with this option. Note it will only work with text files. Enabling this option also activates the Import settings, below.

Sync only documents in collection If this is selected, only documents that are contained in the collection specified in the pop-up list next to this checkbox will be kept in sync with the folder. This essentially acts as a filter for the other two above "Sync" options. For instance, if "Sync the contents of the Draft folder" is selected but "Sync all other text documents" is not, and if this option is selected, then only documents that are contained in the Draft folder and in the specified collection will be synced.

Prefix file names with numbers By default, Scrivener will prefix each filename with a numeral corresponding to its position in the binder. This will keep your files in the same order as the binder, making them easier to find. Disabling this will remove the number and the contents of the folder will be subject to ordinary alphanumeric sorting based on the names of your binder items.

Take snapshots of affected documents before updating Enabled by default. With this option turned on, Scrivener will automatically generate snapshots of each document if it requires updating. If you prefer to handle snapshots manually, you may want to turn this off. However be aware that leaving it on is the safest option, particularly when both items have been accidentally edited separately. Since Scrivener cannot determine which is meant to be the most up-to-date other than by the file modification date, having snapshots available will allow you to review the specific changes and decide how to manage conflicts should they occur.

Check external folder on project open and automatically sync on close Enabled by default. When this project is opened, it will briefly scan the contents of the external sync folder and alert you if there are any changes detected, offering you the ability to update your project immediately. When closing, the same check will be performed. This option will ensure that the sync folder and the project remain up to date, though they will differ while editing, until you run sync again.

The **Import** section allows you to choose where files that have been created in the Notes folder outside of Scrivener will be imported. You can select a target container (Research by default), or uncheck the "Only show containers in destination list" option

to choose any item to become a container for imported files. This section will be disabled if "Sync all other text documents in the project" is turned off, above.

The **Format** section determines how the files will be created in the sync folder. For instance, scriptwriters might not want to use a script format for notes and research documents, so there are separate options for files stored in the Draft folder and other files in the project. There are three file formats available, and depending on your intended purpose, selecting the right option will be important:

- *Plain text (TXT)*: **Formatting will be stripped from edited portions** when using this option. The resulting files will be standard, plain-text files in the UTF-8 format. If they are edited outside of Scrivener, when synced back in, all custom formatting in paragraphs that have been edited will be lost. Scrivener will protect those areas of the files which have not been edited, wherever possible. Scrivener does its best to retain as much formatting as possible when re-syncing, but if formatting is as important to your workflow as synchronising, then you will either need to find a solution that can take advantage of RTF files, or save formatting for the final stages in your writing project.

  When plain text is the chosen format, you can change the default extension that Scrivener will use for the exported files. If you use an editor that displays markup, for example, it can be handy to change the extension to the markup's accepted form, like .md for Markdown. When using this option, be aware that files added to the sync folder using a different extension will not be recognised. This means once you choose an extension you should stick with it (lest all of the existing files become unrecognised and moved to the "Trashed Files" folder). If you wish to change the extension you are using, use a batch rename tool to fix the sync folder, or regenerate it from scratch.

- *Rich Text (RTF)*: This provides the cleanest transfer of information. Most formatting will be retained, especially when used in conjunction with a word processor that handles all of Scrivener's RTF features, such as Word or Nisus Writer. This is the best option for collaborating with other individuals who do not have access to Scrivener, or for working in a multi-platform setting yourself. All versions of Microsoft Word fully support the RTF format.

- *Final Draft (FDX)*: Those working with Final Draft, version 8 or greater, and scriptwriting mode should use this setting, as it will retain all special script formatting in a round-trip.

- *Fountain (.fountain)*: For use with screenplays, this popular plain-text markup format can be edited anywhere plain-text files can be edited, and so has a similar appeal

to Markdown for its portability. Scrivener's internal scriptwriting format will be converted to Fountain syntax when updating the folder contents, and then in reverse the syntax will be converted to Scrivener elements when updating the project. Additional formatting supported by Fountain markup will be converted as well. For details, read more about Working with Fountain (section 20.6).

All files in the Draft folder will be treated as Fountain format for purposes of import and export. This includes any new text documents added to the "Draft" sync folder from outside of Scrivener. If you wish to add notes or comments in the form of new files, enable the ability to sync outside of the Draft folder, using another format than Fountain, and place your text files in the "Notes" sync subfolder.

Automatically convert plain text paragraph spacing When this option is engaged, Scrivener will intelligently convert your documents' paragraph spacing to better suit the intended environment. Since plain-text editors cannot display pseudo-spacing between paragraphs, this option will insert a second carriage return to help set them apart from one another. Upon import, these extra carriage returns will be removed for you.

> **Using Folder Sync with the iPad and iPhone**: If you intend to use this feature in integration with Dropbox and the iPad or iPhone, you should always use the plain-text format. At the time of this writing, there are no Dropbox aware editors on the iOS that can read and write RTF files.

Keep this setting turned off if you require a certain standard, one way or the other, and do not want Scrivener to adjust things for you. In particular, those working in the MultiMarkdown workflow require double-spaced paragraphs, and so will not want Scrivener to remove these spacings upon import. Note that this option only affects plain text (TXT) documents.

When using plain-text, inline annotations and footnotes (chapter 18) will be scanned for within the external files, and any text matching the following patterns will be converted to the corresponding notation style. Using this feature, you can round-trip your notes, and even create new notes in an external editor. Annotations will use double-parentheses, and footnotes will use double-curly-braces.

Additional settings which impact export and import can be found in the application preferences, under Import & Export. If you prefer to work with inline annotations and footnotes, make sure the default setting in this pane has been changed, otherwise your RTF comments will all become Inspector comments after a sync cycle. For further documentation on this pane, see Import & Export Preferences (section B.10).

13.2.2 Usage

After clicking the **Sync** button for the first time, Scrivener will export a copy of every item according to the options above. If the export folder is located in an area where it is accessible to other computers and mobile devices, you can log on to Dropbox with your external application and start browsing the files as soon as they finish uploading. Note that folders in the Scrivener project will appear as ordinary files in the sync folder, because Scrivener folders can contain text.

Scrivener will export all items associated with your manuscript into the **Draft** folder. If you choose to export non-Draft files, everything else will be placed into the **Notes** folder. If you intend to start editing these files immediately, it would be a good idea to close your Scrivener project at that point. While Scrivener will make scrupulous copies of everything it changes (unless you've disabled snapshot generation), it is best to work in an alternating pattern to reduce confusion between which file is the most up to date.

> **Using the sync folders as an inbox**: In addition to creating files for material that already exist in your Binder, you can use this feature to collect notes while you are away from Scrivener. If you intend to use this feature only as an inbox, that is, for collecting new notes and not working with existing material, you can set up sync to use an empty collection. The empty Drafts and Notes folders will be created for you if their respective checkboxes are ticked. Any files you create in this folder will be imported into the project in the future, and will be available for further sync unless you explicitly remove them from the Collection filter.

In a collaboration environment, it may not be possible to wait until the other person is finished. It is safe to work in both the project and the exported copies at once, so long as changed files are looked over after syncing. In most cases Scrivener will select the best option for you, but in cases where both you and your colleague have changed the file in between syncs, you might need to resolve the differences using Snapshots and the Compare feature. For more information on using Snapshots, see the Snapshots Pane (section 19.8).

When performing a folder sync, only resources which have changed (either in the project or on the disk) will be synchronised, to save time. In the case where project binder items have been updated from the disk, a list of these changed items will be presented to you upon completion. The list will not contain items which have only changed on the disk. Use this tool to browse through the changed items and review them with the Snapshot Compare feature. This interface is much like Project Search, and can be dismissed by clicking the **X** button in the bottom corner of the binder footer bar.

For those that want to dig a little deeper, this is really a Collection that will be created and titled "Updated Documents", with each changed project document added to it.

Every time sync needs to change project files, it will look for a Collection named "Updated Documents", and replace the contents of it with the results of the latest sync, so if you wish to retain a list of changed files for future reference, change the name of the Collection to something else.

New files can be created while you are away from your project. Simply name them whatever you would like them to be called in the binder, and save them to the Draft or Notes folder. Scrivener will import files created in the external Draft folder into the project's Draft folder, and files created in the external Notes folder into the designated auxiliary container (Research by default), when you sync. Initially they will be imported at the bottom of their respective containers, but once you sort them to their proper location, they will sync from that point forward in an ordinary fashion.

When you have returned to your computer and wish to integrate any of the changes made, simply load the project in Scrivener. It will, by default, check for any changed files and alert you that your project is out of date. If you have this option disabled, or the project is already open, you can manually activate the sync panel again, using the menu item, and click the **Sync** button. Moments later, your project will be updated with all of your external changes.

While working in Scrivener, you can easily sync without going through the setup process. Use the **File ► Sync ► Sync with External Folder Now** menu command, or add the "Sync" icon to your toolbar. The icon and menu command will be disabled until you first set up sync settings, or if Scrivener detects that the sync folder has become damaged or removed.

13.2.3 Tips for Working with Synced Folders

Here are a few guidelines which, if followed, will prevent problems in everyday use:

- *Alternate between external files and the project*. This is a rule of thumb, not a strict rule. Scrivener reads the modification date of each file, and uses the latest one to determine which should be the binder copy. It does not examine the contents of files. It is thus best to alternate between using the external files, and the project folder, rather than working in both separately for a while, and trying to merge them later.

- **Never try to sync one project's folder with another**. Even if those two projects originally came from the same identical project, over time there will be differences in each project that you cannot see, and these differences will lead to confusing results at best, and a loss of data at worst. A sync folder is meant to allow you to work without Scrivener for a while, and then come back later and update your project with those changes.

- *Disable automatic sync on open and close before duplicating the project in Finder*. If you create a duplicate from within Scrivener (using the back up features, or File ▸ Save As...), you do not need to worry about this as Scrivener will clean up any automated settings for you. However if you intend to duplicate the project outside of Scrivener using Finder, make sure to disable this option first, otherwise they will both end up using the same folder automatically, and this can lead to mangling your project if you continue working in both projects. Once you've created the duplicate, it is safe to turn it back on in the original project, provided you either leave automatic syncing off in the new project, or point it at another folder.

- *When working from multiple computers sharing the same sync folder, always make sure your project file is the most current*. It is perfectly safe to use the same project to sync to a shared folder from multiple computers provided you are always using the most recent version of the project and are not trying to bend the feature to sync two different versions of the project (see above). You will receive a warning dialogue the first time you try on each system, but in this one case it can be ignored.

- *Avoid editing the filename*. The numeric value on the very end of the filename enclosed in hyphens, such as "-78-" (the precise number will be different for each item), is vital to keeping the file identifiable by the item it is linked with in the project; damaging the number could produce unintended duplication, or data loss in some cases[3].

- *Session target goals will include synced changes*. When updating project files with external edits, the session target will be incremented by the amount of text that has been added. If you wish to keep track of your session target separately, you might wish to reset the counter after updating your project.

- *If using mixed inspector comments and inline annotations*: When using sync folders, it is best to use either all Inspector footnotes and comments, or all inline annotations and footnotes, and set your Import & Export preferences accordingly. In some rare cases, comments adjacent to annotations can result in one of the notes getting lost, due to the limitations of the RTF format.

[3] You can safely edit the filename so long as the dashed number is untouched. When a file has been edited in the sync folder, if the binder name portion of the filename is changed, the sync function will update the binder with the new name. This is considered an advanced feature, however, to be used with care. Note that changing the sequence number will have no effect on the binder order. Also note that if all you changed is the filename, nothing will happen due to the way modification dates are stamped on the filesystem. Advanced users could use the UNIX command, touch, to force an update.

> **Dropbox and iOS**: When using the Dropbox app with the iPad or iPhone, you can elect to have documents stored on your device. Ordinarily, you need an active Internet connection to view files, but when a file has been marked as a "Favorite" it will be stored on your device for perusal at any time. Note that if you are using a third-party editor to interface with Dropbox and allow remote editing, this trick will *not* work, but many of these editors store their text files on your device as well, and any changes made to them will be synced the next time you have an Internet connection.

If you are using this feature in conjunction with another author or editor, make sure to communicate these ground rules with them where relevant. Since it is not possible for Scrivener to guess at your intentions, good communication over where new files should go, and what is being independently worked upon will be beneficial.

13.2.4 Limitations

RTF Format Limitations: When using the RTF export format, you can expect zero to minimal loss of formatting information when used in conjunction with a good word processor. There are a few features in Scrivener that have no comparison in RTF, such as multi-colour annotations and comments, and Scrivener links, which will be normalised or lost, respectively.

- If you use sequential annotations separated only by colour, it would be a good idea to move them so that they have a word or two in between them, or separate them by putting them on different lines.
- Expect loss of some features when using RTF editors that do not fully support the RTF specifications, such as TextEdit, Pages and similar. Footnotes, comments, lists, images, and tables are the most common items which have limited to no support. Using fully-featured word processors with good RTF support, such as Word or Nisus Writer Pro will help you avoid this, and even be quite useful as comments can be used to aid in the collaboration process.
- If the only changes made to a file were formatting, they will not be included in the sync. This is down to a technical limitation of how these files must be scanned and processed. So if you have formatting changes you'd like to add, just add a temporary word at the top that you can remove later.

Plain-text Limitations: Since it is impossible to convey formatting in plain-text without some sort of visible mark-up, Scrivener takes steps to protect as much of your formatting as it possibly can. As with RTF, only those files which have been edited will

be transferred back to your project, but further, each changed file will be analysed at the paragraph level and only replace those paragraphs that have been changed, leaving the rest of the file untouched with its original formatting.

- To avoid having inline notation become confused with standard text, Scrivener will export inline annotations by wrapping them in double-parentheses ((and)). Inline footnotes will be similarly wrapped in curly braces {{ and }}. These work in both directions. If you type them into a paragraph using your text editor, they will be converted to their respective type of notation upon syncing.

- Inspector comments and footnotes will not be exported. If the retention of this information is important to you, it is recommended you use inline notation with plain-text. You can easily convert your notes to inline with the **Format ▸ Convert** tools.

- Embedded images will be stripped from documents that have been edited outside of Scrivener.

13.2.5 Disabling Synchronisation

If you no longer need to maintain an external sync folder, you can shut off this feature by clicking the **Clear** button in the setup screen, alongside the chosen path. At that point the "Sync" button will be removed and replaced with a "Save" button, allowing you to confirm this window without actually attempting to sync. Additionally, removing the sync folder from your drive, moving it to another location, or changing its name will automatically cancel synchronisation the next time synchronisation is attempted.

13.3 Which to Use?

If you have no investment in either Dropbox or Simplenote, you might be wondering which is the best to use. Simplenote is designed to be an easy way to work with note files. When you load the app on your mobile device, it automatically hooks into the central Simplenote server if possible, so as you work your changes are saved globally and all of your other devices are updated with the new notes and text. Consequently, Simplenote is very easy to use. There is no configuration you have to do, you simply download their free app from the App Store and start working with the scenes you sent out for sync from Scrivener.

If you are looking for a quick and painless way to edit your manuscript on the go or easily add thoughts and ideas to your projects, then Simplenote is an excellent way of doing so.

Another advantage to Simplenote is that it uses a central server instead of your hard drive. This can be particularly important if you use more than one computer, and not all of your computers have the same log in name. Syncing with an external folder requires the names of places to be the same no matter what you computer you are on—and this can be impossible in some cases without complicated work-arounds. Simplenote doesn't care if your user name is one thing or another, so it will always work wherever you go.

While much has been stressed over the simplicity of Simplenote, this should not imply that Dropbox is complicated to use. It is, especially considering what it is capable of doing, very straight-forward and useful within minutes of installation. There is much you can do with Dropbox, and most of these topics would be out of scope for this manual. Specifically in terms of Scrivener, using a sync folder with Dropbox gives you extra power because you have more control over what you use to edit the files. Unlike Simplenote, everything is stored as simple text files on the hard drive of every computer or mobile device you install Dropbox on. That means you can access them just as you would any file on your computer, with any program you like.

If you need to do more than one thing with your manuscript, outside of Scrivener; if you like to choose the apps you use to edit text; if you want to keep all of your external notes easily accessible as ordinary files, then Dropbox might be your best bet.

13.4 Index Card for iOS

Index Card for the iPad and iPhone is a corkboard and outlining app that was partly inspired by Scrivener. It allows you to create projects comprising index cards (each of which holds a title, synopsis, and main text, just like Scrivener's index cards) that can be moved around and edited on a virtual corkboard, or in a list view resembling Scrivener's outliner. It is therefore an ideal companion application for those on the go, and using Scrivener's "Sync with Index Card for iOS" feature you can package up parts of your Scrivener project for editing and restructuring in Index Card on your device.

The synchronisation process involves creating a collection in Scrivener, saving it as an .indexcard file that can be read by Index Card, and later updating the collection in Scrivener from the updated .indexcard file written out by Index Card after you have made changes on the go. There are therefore four steps involved in syncing with Index Card:

1. Creating a Collection to hold the documents to sync.
2. Saving an .indexcard file to your Dropbox folder.
3. Opening, editing and saving the .indexcard file in Index Card on your device.

4. Updating the Collection in Scrivener with any changes you have made.

Because syncing with Index Card relies on Dropbox, which is a service that allows you to sync files between different devices that has a free option. **Before you begin, you will need to ensure that you have Dropbox installed on your computer**. You can download the desktop version of Dropbox, here[4].

You will of course also need Index Card[5] installed on your iPhone or iPad.

13.4.1 Creating a Collection

Index Card projects consist of a single stack of index cards that can be moved around on a corkboard. Because Index Card expects a flat list of cards, it is not possible to sync the entire structure of a Scrivener project, or even all of the contents of the Draft folder, which will most likely contain documents inside folders several levels deep.

For this reason, the Collections feature is used to sync with Index Card. The Collections feature allows you to maintain different flat lists of any documents in the project, which can be ordered however you wish. You should thus create a collection before you sync with Index Card. After editing your project in Index Card, you can easily assign your new arrangement of documents to the binder.

To create a collection, first select the documents you wish to take to Index Card in the binder. Next, click on the "Collections" icon in the toolbar to reveal the collections pane if it is not already visible (or select View ► Collections ► Show Collections). Finally, click on the + button in the collections header bar.

A new collection will be created that contains the documents you had selected. You can rename the collection to anything you like, but something meaningful such as "Index Card Sync" might be a good idea. (You can change the colour of the collection by double-clicking on the colour chip in its tab when the collection is active.)

> **Adding Whole Folders To Collections**: If you wish to include the contents of an entire folder in the collection, select the folder and hold down the Option key down while clicking on the "Add Collection" button.

To add other documents to the your collection, you can just drag them from the binder onto the collection tab.

Read Using Collections (section 8.4) for more information on collections.

[4] http://www.getdropbox.com
[5] http://www.denvog.com/iphone/IndexCard

13.4.2 Saving the Index Card File to Dropbox

We have now set up a collection ready for syncing with Index Card. The next step is to create a file in our Dropbox folder that Index Card will be able to read. To do so, use File ► Sync ► with Index Card for iOS...

Choose the collection you created in the first step from the "Collection" pop-up button. There are three options you can select from:

Sync includes main text and notes This feature only works with newer version of Index Card, which support adding notes and main text to index cards, as well as synopsis. With this option, Scrivener will store the main text of your draft for that section into the text field for the card, and any document notes into the corresponding notes field in Index Card. This way you can not only see your cards and work with them, you can edit their contents too.

Do take notice that by default Index Card will not include notes in an exported .indexcard file. If that option is left off, no changes you make on the iPad to the main text will be updated in the project.

Take snapshot before updating main text As with the other synchronisation tools, leaving this option on will automatically create a snapshot of the item prior to changing its text contents, providing an extra layer of safety.

Convert plain text paragraph spacing If you have formatted your paragraphs with single breaks between them, this will optimise paragraph spacing to look good in a plain text file (which Index Card needs for the notes field). Likewise, paragraphs will be converted back to single breaks when updating the main text.

This option is not necessary (and indeed would be undesirable in most cases) if the source .scriv project already uses double-breaks between paragraphs.

Once you have everything set up, click on Create or Update Index Card File....

The "Save As" panel will appear, and it should automatically point to the "IndexCard" folder inside your Dropbox folder. Index Card creates the "IndexCard" subfolder in Dropbox whenever you sync, so if you haven't saved any files from Index Card yet, it may not exist. If it doesn't, create it yourself—navigate to your Dropbox folder, click on "New Folder", and name the new folder "IndexCard" (with no space). You will need to use this folder, otherwise Index Card will not see your exported files.

Choose a name for the .indexcard file (or use the default one that has been entered automatically) and click on "Save". At this point, an .indexcard file will be saved into the Dropbox/IndexCard folder, and the sync sheet will close. You are now ready to carry your work around with you.

13.4.3 Editing in the Project on your Device

To open the .indexcard file on your iPad or iPhone, follow these instructions:

1. Open the Index Card app on your iOS device
2. If you haven't done so already, tap on the 'gear' button and link your Dropbox account to Index Card
3. Tap on the 'gear' icon from the main project list and select **Copy from Dropbox**
4. Select the .indexcard file that you created in Scrivener (the title of the Index Card project will be the title of your Scrivener project with the name of the collection you synced in brackets). **If you do not see it here**, you need to make sure and save it to the "IndexCard" folder in your main Dropbox folder.

You should now be see the .indexcard file you loaded listed in the main project list. The title of the Index Card project will be the title of your Scrivener project with the name of the collection you synced in brackets.

You can now move the cards around, edit their titles and synopses, and create new cards, just as you would in any Index Card project.

13.4.4 Updating the Collection in Scrivener

When you get back to your computer, you'll want to update your Scrivener project with the changes you have made in Index Card. To do this, you'll first need to update the .indexcard file in the Dropbox/IndexCard folder, as follows.

> **Important Note for Index Card 4 for iPad**: Scrivener will not sync with the newer Index Card 4 version format. This is not a problem, however, as that version has the ability to save as a version 3 .indexcard file. Use this option when transferring your changes back from the device to your original project on the Mac.

1. Click on the "Send" button in Index Card and select "Copy to Dropbox" from the list that appears, using the .indexcard format option. If you are using Index Card 4 for iPad, then choose to export as version 3 .indexcard format.
2. If you are syncing main text as Index Card notes, then make sure the "Include Notes in Export" option is enabled at this stage, otherwise your text changes will not be included.

3. At this point, Index Card, detecting that the file already exists, may ask you whether you wish to overwrite the existing file or save using a different file name. Choose to overwrite the existing file, if necessary.

4. Now, back in Scrivener, go to **File ► Sync ► with Index Card for iOS...** again. The collection we chose to sync should already be selected for us in the pop-up button.

5. This time, choose "Update Collection from Index Card File...", because we want to update our project with the changes we made in Index Card.

6. The standard Open panel will appear, and the .indexcard file that you edited should already be selected for you. If not, select it. Then click "Open".

The collection in your Scrivener project should immediately be updated to reflect the changes you made on your mobile device—the order will be changed if you moved things around, the titles and synopses of any documents you edited will be updated, and any new index cards you created will appear as new documents in the collection. Additionally, when you return to the binder, you will find any new documents that were created have been placed in a folder below the Research folder, which will be named using the title of the collection followed by "Unsorted" in parentheses.

If you deleted any index cards in Index Card on the device, they will be removed from the collection in Scrivener but will *not* be deleted from the binder.

Committing Rearrangements to the Binder

Collections exist in parallel to the binder, allowing you to gather disparate documents from all over the binder in arbitrary lists in any order you wish. If you've spent a lot of time in Index Card on the iPad rearranging the cards and finding their most effective order, at some point you will want to take that arrangement to the binder rather than having it sectioned off in a collection. This is easy to do, and there are several ways of doing it. Here are the easiest:

With the collection visible, select all of the documents in the collection list. Then, either:

- Ctrl-click on the selection, select **Move To** from the contextual menu that appears, and then choose a folder into which to move the documents. When you do so, the binder will reappear to show the documents in their new location.

- Drag the selected documents onto the "Binder" tab in the Collections pane and hold them there. After a second, the collections list will disappear to be replaced by the binder, and you can drag and drop the documents wherever you want to place them.

Note that in both cases, the collection remains unaffected, and continues to exist separately.

When you next want to take your work with you on the iPad, once more ensure that the collection contains the documents you want to take, and go through the process again, overwriting the old .indexcard file with one generated from your updated collection.

> **Syncing Multiple Collections**: It is possible to sync more than one collection in a project with Index Card—each collection will appear in Index Card as a separate project. However, bear in mind that if any documents appear in more than one collection, any edits you make to them in one Index Card project could get overridden were you to later sync with an Index Card project created from another collection that contains some of the same documents.

Also note that Scrivener will warn you if you try to sync with an Index Card file that was not created with the selected collection.

13.5 Scrivener Everywhere

All of the methods discussed so far have worked on the assumption that you only need to work on a specific aspect of your project when away from home, but many people prefer to work in several places, and like to use a second computer like a laptop. Many others would prefer to have total editorial control of the likes only Scrivener can provide, rather than simply working with content or high-level organisation in an index card style application. Keeping more than one computer synchronised can be a lot of work, but keeping one Dropbox folder (or similar service) synchronised between computers requires little thought at all.

It therefore follows that placing your Scrivener project into a Dropbox folder so that it is immediately available to every computer you own, would seem a perfectly natural progression. What you should be cautious of, when entertaining this idea, is that a Scrivener project is comprised of many smaller files, and that they all need to be working together as a cohesive whole. Ordinarily, this is not a problem. Nothing goes into your project and messes with it, except for Scrivener. When you place a project on Dropbox, however, you are giving Dropbox permission to scrutinise and curate the individual pieces of your project that ordinarily only Scrivener messes with. This goes for most synchronisation technologies in general, but since we've tested Dropbox the most, this section will be framed with that service in mind.

By and large, this is safe to do. Dropbox is designed to carefully make sure each computer has a thorough and exact replicate of what all your other computers have. However,

a problem can arise when one computer edits your project in a way which conflicts with the copies on the other computers. This can be easy to accidentally do. All it takes is working on a copy of a project that is not yet fully synced. This might happen because your computer has been offline for a while, or because Dropbox never got a chance to fully update all of the files in the project.

When this happens, copies of each "conflicted" file will be duplicated with different names so that you can decide which one is the best copy, or manually merge them together in a text editor. Ordinarily, this is fine and easy to do, but inside of a Scrivener project, it can be more complicated. For one thing, the full contents of your project might not even be easily visible. Secondly, some of the files in a project are very technical in nature, and if you are not fluent in XML (not to mention Scrivener's schema), it might be impossible to best decide which version is better. The solution is to never get into this position in the first place, and this is fortunately easy to do.

To avoid conflicts with synced projects:

1. **Always make sure your syncing software is done syncing before you open a project**. Good syncing software, like Dropbox, will let you know when it is moving data to or from the central server on the Internet. In the case of Dropbox, a small icon will be placed into the status menu bar area, in the top-right portion of your screen across from where the main menus are. Keep an eye on this indicator and wait for the green checkmark[6] to appear before you open the project.

2. **Never open a project more than once**. If all goes well, Scrivener will warn you if you try to do this, but in some scenarios this warning might fail, so try to always remember to close your project when you are done with it. Ignoring this rule is a surefire way to cause major headaches.

3. **Always wait for your syncing software to finish syncing before you shut down your computer**. This is of course a corollary to the first rule. Just as you should wait for your computer to be updated before opening a project, you need to make sure that all of your changes made to the project have finished syncing back to the server (and thus made available to your other computers), before you put it to sleep or shut it down.

4. **Reduce auto-save frequency**: this step is optional unless you are working on a very slow connection, such as dial-up. Editing a Scrivener project can generate a

[6] It may be possible to customise the status icon, and not all of the visual themes use a green checkbox to indicate when Dropbox is in standby mode. Familiarise yourself with how your preferred status icon works.

lot of network traffic because of how often Scrivener saves files to the disk. With modern high-speed connections, this should pose no problem, but with slower or unreliable connections to the Internet, it's a good idea to set Auto-Save (in General preferences) a little higher so that changes are sent in larger batches to the Dropbox server, rather than trickling in constantly. When working this way, you might want to save manually more often, with the File ► Save menu command. **Alternately, pause Dropbox**: the newer versions of Dropbox allow you to pause syncing. This can be done from the status icon as all other functions with Dropbox can be accessed. Pausing syncing means that all changes made to the Dropbox folder will not be immediately synced to the server. Thus there is no need to mess with changing the auto-save frequency, and it keeps your entire working session in one "version" within the Dropbox system. Of course, you will need to remember to resume syncing prior to shutting down the system.

If you follow these three simple rules, you will dramatically decrease the chance of any strange conflicts arising due to working in a live syncing folder with Scrivener. As with all synchronisation technology, it is inherently "less safe" than working local and saving remote. It is possible to work safely for years in this fashion, but extra vigilance will be required of you and everyone participating in the shared project.

While this section has primarily focussed on Dropbox as a means for keeping your project files synchronised between computers. There are many technologies for doing so these days. If you have another solution you are using and prefer, test its viability with copies of your project, or test projects, prior to committing important work to it.

A note on using such sharing services for collaboration: while it is possible for multiple parties to work off of a single .scriv file hosted on a shared network folder, extra caution will need to be exercised in order to keep the project data safe. At the very least, this section of the manual should be distributed to all parties concerned, and good lines of communication should be established so that everyone knows when a person is actively working in a project. There are fail-safes programmed into Scrivener to let you know when another session is currently active in a project, but making sure everyone is on the same page with who is editing the project will ensure no mistakes are made. Guideline #2 above deserves to be reiterated: **never open a project for editing if another person is currently editing the project**.

To date, only one such technology is known to be a risk to use in this fashion, and that is Google's Drive, which simply is not designed with complex formats in mind. It is not recommended that you use Google Drive for live projects under any circumstances. It is however, quite safe to use it to "transport" zipped backups of your projects from one computer to another.

Finally, it has been said before, but it bears repeating: Scrivener's built-in folder sync tools should never be used to keep two projects in sync. They are not full projects, and because of their simple and accessible nature, do not contain nearly enough information to properly inform a project of everything it needs to keep your data safe, but rather only what they need to interface with *one* single project.

Part III

Writing

> Writing is easy. All you do is stare at a blank sheet of paper until drops of blood form on your forehead.
>
> Gene Fowler

When it comes to the process of writing itself, nobody can tell you how it should be done. Each author has their own methods, their own rituals, and their own favourite tactics. Scrivener was designed to recognise that everyone is different, and as a result the program features an extraordinary amount of flexibility and interface power. You will find workflows and tools for all manner of writing projects, from a doctoral thesis, the next blockbuster screenplay, to a novel, game design, a collaborative scientific article, patent claims, biographies, and much more. Because of this, there are many features you just won't need! That is fine, because Scrivener has also been designed to keep these features out of your way unless you need them.

Consequently, this section contains chapters which could be considered optional. In particular, screenwriting and bibliographies can be safely skipped unless your works require these functions.

The exception are the chapters discussing The Editor, and Writing and Editing. These will be of benefit to all, as this will introduce the text editor, where you will be doing most of your writing, and if you haven't already done so, at least skim the prior part on preparation. Vital concepts have been introduced there, and many of the following chapters will build off of those concepts.

The topics that will be covered in this part are:

- The Editor (chapter 14): Full documentation of the various editor views and how to use the primary editing and viewing interface.
- Writing and Editing (chapter 15): Formatting, layout, and tools to enhance the editing and revision processes.
- Page View (chapter 16): An alternate writing environment that emulates the printed page. Useful for workflows which revolve around page counts, or just as an aesthetic way of watching your work build pages, instead of a single scrolling view.
- Composition Mode (chapter 17): Scrivener's composition mode feature. Wipe aside all distractions and just *write*!

- Annotations and Footnotes (chapter 18): Learn how to comment as you write using either Scrivener's exclusive inline annotation tool, or linked comments. Also covers footnotes/endnotes; marker highlighting; and revision pens.
- Inspector (chapter 19): All about the inspector, which displays document metadata, notation, references, and more.
- Scriptwriting (chapter 20): While Scrivener is not meant to replace an industrial strength scriptwriting tool, it can still function as a very adept first-draft scriptwriting solution, and produce exports which will work seamlessly in industry standard applications, like Final Draft.
- Writing Tools (chapter 21): General tools for writing and using the software more effectively, include:
  - Searching and Replacing (section 21.1): Comprehensive tools for finding items, words, notations, and even formatting are discussed here.
  - QuickReference (section 21.2): Taking Quick Look to the next level. Open items in persistent, editable windows. When two splits aren't enough, use QuickReferences.
  - Goals and Statistics Tracking Tools (section 21.3): Scrivener has numerous tools for analysing the content of your project, as well as tracking goals and progression while you work.
  - The Name Generator (section 21.5): Create fictional names with ease, using the name generator.
  - Using Equations with MathType (section 21.7): If you need to insert formulae into your work, Scrivener integrates with the popular MathType software.
- Using MultiMarkdown (chapter 22): If you prefer to write in a structured or semantic style rather than with a rich text focus, MultiMarkdown provides a simple, easy to read way of defining and using styles which can be exported to word processors, web pages, and LaTeX.

Chapter 14

The Editor

14.1 Header Bar

The header bar appears at the top of each editor pane, and can be toggled on and off via **View ▶ Layout ▶ Show (Hide) Header View**. It contains useful navigation tools, displays the current title of the edited document(s), and provides control over editor splits.

The header bar has three colour states:

1. Grey: when no split is active, this is the default state. If splits are turned on, the inactive editor will be grey.
2. Blue (or graphite when the dark grey theme is enabled in the Appearance preference panel): when splits are active, this denotes the *active* split. Any actions taken which will impact the current editor will be made to the blue document. The title itself will also be underlined for the active split.
3. Red: when an editor has been locked, it will turn red to remind you that clicks in the Binder will not load any files into the editor.

14.1.1 History

On the left-most side of the header bar are history navigation arrows, which should be familiar in appearance and usage from any web browser. Use these to navigate backward and forward through the navigation history. If you right-click on the either of the arrow buttons, a menu will be provided, making it easy to jump straight to something far back or forward in the history without having to go through each point in between.

14.1.2 Item Icon Menu

Next to the navigation buttons is the document icon, which shows what type of document is being displayed (which can sometimes be abstract, if more than one item is selected for display in the view). Clicking on this icon brings up a menu containing some useful shortcuts to oft used commands. The icon menu works on the current *session*, which can at times mean more than one file, when working in Scrivenings mode.

Reveal in Binder (⌥⌘-R) Will show the location of the currently edited file in the binder, opening any containers as necessary to do so. This is most useful when the method you used to arrive at the current document did not involve clicking in the binder (such as using the history navigation buttons or a such result), or if you are currently viewing a collection and wish to find where the file is actually located in your project outline.

When used with Scrivenings mode, all of the entries included in the session will be highlighted in the Binder.

Path Reveals the location of the current document in descending order. The top entry will always be the current document; the entry below that its immediate parent; and so on until the top of the project binder is reached. These can be selected to immediately jump to any ancestor of the current item. This command always works on individual files. When a Scrivenings session is in use, the constituent portion of the session that you are currently working on will be revealed in the path.

Go To Offers an alternate way to jump to a specific binder item without using the binder. If the binder is hidden or you would rather not scroll and open folders, for instance, it will let you navigate to other areas of the project without having to alter your layout. Your "Favorites" (section 9.6) will be placed in the top of this menu, making it easy to jump to them from anywhere in the project.

When viewing a Scrivenings session, the contents of this menu will operate within the confines of the current session only, thus providing a miniature "table of contents" for the current session.

Bookmarks Displays any text bookmarks (section 15.8) that have been added to the current document in a handy menu. Header style bookmarks will be offset and set to bold, making it easy to cut a larger document into sections of bookmarks. If you are viewing a Scrivenings session, all of the bookmarks in the entire session will be displayed together.

Bookmarks can be added to your document using ⇧⌘-B, or from the **Edit ▸ Insert ▸** sub-menu.

Take Snapshot This command will take a snapshot (section 15.6) of the current text and store it for later use. This command always impacts only one file. When used in a Scrivenings session, only the constituent portion of the session your cursor is currently in will be saved to a snapshot.

Lock Group View Mode When viewing an container or collection group, this command will be activated. It will lock the currently used view mode for this container, so that no matter what you change it to at a later time, clicking on this container will always return to the selected view mode (subsection 9.4.1).

Match Split Document(s) Will open the currently viewed item in the inactive split. This command will not be available if no splits are open.

Lock In Place (⇧⌘–L) Locks the editor (or split) so that no Binder clicks will affect it. When an editor is locked, its header bar will turn a shade of red (subsection 9.4.2).

14.1.3 Header Bar Title

To the right of the history navigation buttons, the title of the currently viewed item will be displayed. As you navigate through the project, this title will update to reflect the current contents of the editor. In most simple cases, it will display the title of the document you are currently viewing or editing. Whenever the presented title consists of only the *title* itself, you may edit the name of it by clicking within the text, and pressing Enter to confirm. MAC:When a single item is shown, hovering over the title will display the "path" of the item in a tooltip. This is similar to a file path on your system. It will show all of the folders the item came from. You can also view this information graphically in the header bar icon menu, as already described.

When you are viewing a collection of items in a corkboard or outliner, the name of the viewed container will be displayed. That is, if you click on a folder in the binder, the name of that folder will be displayed in the header bar, regardless of which index card or outliner row you have currently selected. Use the Inspector to view the details of selected items within a group view.

When viewing the result of multiple item selections in the binder (when more than one item is selected via Shift-click or Cmd-click), the view will display the title, "Multiple Selection". This is a unique view which lets you know that the corkboard or outliner you are viewing doesn't "exist" anywhere in the project. You are viewing a selection of items, but you will not be able to add new items or re-arrange existing items because there relationship amongst each other does not correlate with anything in the Binder.

In the case of a Scrivenings session that comes about as the result of clicking on a container, the name of the container will be displayed first followed by "(Composite)". Suffixed to this will be the name of the current document you are editing *within* that session. When the Scrivenings session is the result of a Multiple Selection, it will be stated as such, and the name of the currently edited document within the session will be suffixed to that reminder.

In all cases where the title is a not a literal binder title, you will lose the ability to edit the name of the document. You will need to use the inspector or the binder to make this change, or enable **Format ► Options ► Show Titles in Scrivenings**.

14.1.4 Sidebar Navigation

Moving to the right-hand side of the header bar, you will find two arrows pointing up and down. These arrows allow you to navigate within the project sidebar (be it the binder, Search Results, or Collection that is currently viewed), as though it were a flat list. This form of movement is unique in that the binder selection will update. In most cases, when you navigate to documents using the menus or the various controls available in the header bar, the binder selection will not update to follow your select, by design. In this case however, these two buttons are actually synonymous with clicking on documents in the sidebar.

Clicking the up-arrow will move you to the document immediately above the current document in the sidebar. The selected document will always be the one immediately above, even if that item is currently hidden and on an entirely different hierarchical level. Clicking the down-arrow will always move you to the document immediately below.

If you are viewing a container as a Scrivenings session, the behaviour of this feature will modify slightly. The nature of selection will still be the same in terms of linear above/below selection, but the current session will not be dismissed. It thus can be used as a form of navigation within a Scrivenings session. Note that if you navigate above or below the boundary of the selection, it will be dismissed. This feature is an augmentation of the basic behaviour. It is not available to multiple selection based sessions. If you wish to navigate within a multiple-selection based Scrivenings session, use the item icon menu "Go To" instead.

These two buttons can also be activated with the keyboard shortcuts, **⌥⌘–UpArrow** and **⌥⌘–DownArrow**, respectively.

14.1.5 Split View Button

Finally, the last button on the right-hand side is a multi-purpose button which changes depending upon the current layout of the application and your last preferred split orientation.

If no splits are open, a small line will be drawn through the icon in the horizontal or vertical orientation to let you know which way the editor will be split if you click on the icon. If you hold down the Option key, this will switch to the opposing orientation, and that preference will be saved for future splitting as well.

Once a split is already open, the icon will appear blank. Clicking on it in this state will close the *opposing* split. This means you should click on the icon in the header bar

for the document view you wish to retain once the split has closed. Either side can be dismissed by clicking on the opposing header bar button. Additionally, if you hold down the Option key while a split is open, a vertical or horizontal line will be drawn, allowing you to switch split orientations without otherwise disturbing your layout. Both splits will remain open, only the orientation will change.

See also: Splitting the Editor (section 14.4).

14.2 Footer Bar

Below each editor pane is the footer bar. As with the header bar, its visibility can be toggled with **View ► Layout ► Show (Hide) Footer View**. This is the most dynamic part of the editor, in that it will change depending on the type of document visible and the current editor mode. When a text document is being viewed, for instance, it will display the word and character count along with a drop-down menu for changing the text scale and a button for setting document targets. In script mode, it will display scripting hints and the elements menu. When media is displayed, it shows the current playing time. This section will cover the tools available to the standard text editor.

- For information on the various other document footer bars, see Viewing Media in the Editor (section 14.3)
- For details on how to use the footer bar in script mode, see Scriptwriting (chapter 20).
- Read the chapters on The Corkboard (section 12.1) and The Outliner (section 12.2) for details on how to use the footer bar in those view modes.

14.2.1 Text Zoom

You can make the text in the main editor larger or smaller without changing the font by using the text scaling pop-up in the left side of the footer bar. See Scaling Text (subsection 15.3.1) for more information.

14.2.2 Quick Text Statistics

In the middle of the footer, the current word and character account of the text you are editing will be displayed in real-time as you type. This counter works for all visible text, even if it would otherwise not be compiled, and will aggregate all text together when using Scrivenings view. This counter will also display the statistics for the current *text selection* if one exists. When the counter is displaying selected text, the colour will turn blue to indicate that it is no longer counting the entire document(s) text.

Clicking in the statistics area will reveal a pop-over, which will display a little additional information, but more importantly, provides a quick count without footnotes or annotations—though you can selectively turn either of these back on. The live counter that updates while you type will always include both of these for performance reasons, so if you need to get a quick statistic scan without them, use the pop-over. The pop-over also works with a selected range of text.

> **Pro Tip**: If you want to easy grab the word count for the current section you are working on within a Scrivenings session, you can use the selection counting feature mentioned above in conjunction with the **Edit ▸ Select Current Text** command, which can be enabled by holding down the Option key, or pressing **⌥⌘–A**. This will quickly select only the text of the current section you are editing, showing the word and character count in the footer.

14.2.3 Text Goals

When you are editing a single document in standard (not scriptwriting) mode, a small target icon will appear on the right side of the footer bar. This button brings up the target options for the current document which will let you set the numeric word or character goal you intend to achieve with the section. You can optionally use Growl notification to let you know when you've reached your goal. These notifications work in both ways, whether you are working up to a number while writing, or working down to a number while editing.

When enabled, the footer bar will acquire a progress bar next to the target icon which will fill as you type, gradually changing colour until you reach your quota. These targets are set individually for each document. You may also monitor and set them using the Outliner, by revealing the "Target", "Target Type", and/or "Progress" columns, while the "Total Goal" and "Total Progress" columns will show an aggregated goal count and total progress toward that sum, for the displayed Outliner row and all of its descendant items. In this way, you can easily work toward chapter or other larger section goals, while still maintaining a fine-grained approach to cutting up the section into smaller pieces.

To disable the goal metre for a document, enter a value of "0" into the target configuration for the document.

14.3 Viewing Media in the Editor

The editor is capable of viewing most of the file types that you will need to interface with for research and creative use. The footer bar will often change to provide tools for working with different media types. Note that Scrivener is, by and large, not intended to operate as an editor for all types of files. It would take years to design these features,

and in most cases your existing applications will do a better job of editing PDFs, movies, and images. Consequently, a convenience button will be provided on the far-right side of the footer bar for most media views. You may also right-click on any media file in the binder and opt to **Open ▸ in External Editor**, which will open the file using your system default for that type of file, making it simple to load the document in an external editor of your choice. Changes made to the file will be saved within the Scrivener project when you return[1].

14.3.1 Viewing Images

The image view is displayed whenever an image document is selected, allowing you to zoom in and out and rotate it. If the image is larger than the viewable area, click and hold the mouse on the image and move the mouse to drag-scroll the image around. To zoom in and out and rotate, double-click anywhere within the image or the grey area around it. This will bring up the image tools pop-up.

The slider at the top of the pop-up allows you to scale the image.The button to the left of the slider zooms the image as far out as it will go (that is, makes it as small as possible) and sets the slider to the far left; the button on the right of the slider zooms the image as far in as it will go (makes it as large as possible) and sets the slider to the far right. Below the slider, you can check a "Scale to fit" option which will disable the slider and always size the image according to how big the editor is.

There are three additional tools available: there is a button for rotating the image (rotating it 90 degrees clockwise, or anti-clockwise when Option-clicked); a button for fitting the image to the current viewable area; and a button that returns the image to 100% magnification.

In the footer bar, click the toolbox button to open the image in your default external editor. If you wish to open the file in another program, right-click on this button to select an alternate from a menu. This menu is provided by the operating system, so if your favourite program isn't showing it, you can resolve this by sorting out why it isn't showing up in the OS (making sure it is in the Applications folder will solve most issues of this nature). When you select a non-default application to load the item in, Scrivener will ask whether you wish to always open this file in that editor. This will impact *all* files of the same type, when loaded from within Scrivener (it will not change your system default).

Beside this button you'll see a reload button. You will need to click this after saving your edits in the external image editor in order to see the changes.

[1] If you are already viewing that file in the editor, you may need to refresh its view in order to see the changes.

14.3.2 Viewing PDFs

The PDF view is displayed when a PDF document is selected. Its main view looks and acts exactly the same as Preview.app. You can control the display of the PDF document via the **View ▶ Media ▶ PDF Display ▶** sub-menu, and you can zoom in and out of it using the "Zoom In" and "Zoom Out" controls in the **View ▶ Zoom ▶** sub-menu. You can also access many of these tools via right-click on the PDF display itself.

The footer bar contains arrow buttons on the left, which allow you to browse back and forth through the pages of the PDF document, and text in the centre showing the number of pages the PDF document has and which page you are currently viewing.

> **Annotating PDFs**: As with ordinary text, you can select ranges of text in a PDF file and highlight them with ⇧⌘−H. This tool is limited to the yellow colour. You may also strike out portions of text using ⇧⌘−-. It is not possible to insert comments into a PDF using Scrivener. To give a PDF file advanced treatment, view it in an external editor, such as Preview, or Adobe Acrobat. Any changes made to the PDF, once saved, will show up in Scrivener after you refresh the file display.
>
> Note that Skim, a popular PDF viewer, uses its own system for annotation which does not show up in Scrivener and most other viewers. Consult the Skim documentation for details on how to save embedded annotations.

As with images, you can open PDFs in external editors for annotation or more full-featured browsing, and use the reload button to load PDFs from the disk if changes are not showing up immediately.

14.3.3 Viewing Multimedia Documents

The QuickTime view is used to display movie and sound files. From the control bar beneath the viewing area (above the footer bar), you can play or pause the movie, change the volume, or step backwards and forwards through it. In the footer bar, you'll see a rewind symbol with a pause button in the middle of it. This is an optional mode which will show up as blue if engaged, that will rewind your media by a set number of seconds whenever you pause. This interval can be configured in the Navigation preferences tab (section B.6).

To further aid in transcribing, you can use the ⌘−Return shortcut key to pause and resume a media stream even while you are typing in the other editor, never having to leave the keyboard. With the focus is in the media editor, you may skip forward and backward by a number of seconds with ⇧⌥⌘−) and ⇧⌥⌘−(, respectively.

14.3.4 Viewing Web Pages

The web view displays imported web pages. Note that clicking on any links in a page displayed in the web view will open the link in the system's default internet browser. Scrivener does not allow internal browsing simply because this would mean integrating many of the features of a browser into the web view (such as back and forward buttons, a refresh button, maybe even a download manager…) and this is outside of Scrivener's scope.

When the web view is shown, the content of the footer will display the original URL a as a link that can be clicked on to open the original page in the system's default browser.

You can increase and decrease the size of the web page font using the standard View ► Zoom ► sub-menu.

> **Pro Tip**: If you archive search forms, such as http://www.duckduckgo.com/, and searches that you perform will have their results opened up in your web browser. Thus it can be useful to store a number of common research portals in your Binder for quick access. Note that since this uses an archived version of the web page, if the host site changes their searching protocol, your archived search might break (this happens rarely).

14.3.5 Viewing Unsupported Document Types

By default, Scrivener will let you import files into the Binder that it cannot display in the editor. You can restrict this by checking off "Import supported file types only", in the Import & Export preference tab (section B.10). Unsupported file types will use the default system icon.

When loaded in an editor, the file icon and its filename will be displayed. Double-click on the icon to load the file in its default external editor, or right-click on the toolbox icon to select an alternate application (and optionally set it as the default for that file type). The toolbar will also feature a Quick Look icon. Clicking this will produce the same effect as using Quick Look in the Finder.

When unsupported file types are viewed in the editor the document icon will be displayed. To view the content:

- Double-Click the icon to load the file in the default external editor for that file type.
- Click the "Open in External Editor" button in the editor footer bar, or right-click on the toolbox icon to select an alternate application (and optionally set it as the default for that file type).

❧ Use the **Documents ► Open ► Open in External Editor** menu command (^⌘–O).

As with supported files, unsupported files are fully imported (copied) into the project package itself.

14.4 Splitting the Editor

Scrivener's editor uses a technique known as editor splitting. You may have encountered split screen editing in other applications, but the level of integration and power between the two splits in Scrivener is likely to be unfamiliar to new users. Rather than arbitrarily splitting the interface any number of times and ways, the editor uses a two-way split system so that you can easily control the flow of information in the opposing split while you work.

> **Splitting Documents**: Looking for tips on how to permanently splitting a document into two pieces? Scrivener supports that ability as well, so if that is what you are looking for, you should head on over to Splitting The Document (section 15.3.3) for further details. Splitting the editor is a purely visual tool.

If you are unfamiliar with splitting an editor in general, you can think of it as a way of opening a second editor in which you can view the same item you are already working on with an independent scroll position, or another item entirely. When the same item is loaded into both splits (which is what happens when you create a new split), any edits made to either side will be immediately shown in the other split (in a text editor, your typing will be mirrored; with a corkboard, cards you add or delete will show up on the other side too). Using two splits makes it easy to edit or refer to other parts of the document, or your research & notes, without scrolling back and forth or frequently use the history buttons.

Loading another item entirely into the second split makes reference a snap. While you could use the history functions to jump back and forth between a reference source and the portion of your manuscript that you are editing, splits will present both items to you at once, even allowing you to play media in the second split while recording notes or transcribing it as it plays.

In addition to standard file viewing and editing, you can also use splits to do anything else that you would ordinarily do in a single editor. You can mix Corkboard, Outliner, and Scrivening sessions together, and even link the splits so that clicking on cards or outliner rows automatically loads the item in the second split for you.

We'll now take a look at all the available possibilities in turn.

14.4.1 Horizontal and Vertical Splits

The editor interface can be split in one of two orientations. In vertical mode, the divider will be drawn down the middle of the screen, and is most suitable to viewing two text documents side-by-side as it maximises vertical space. Horizontal mode creates a split that is familiar to many applications like Mail with content on the top and bottom. In either case, both sides of the split have the same degree of power in loading various views, media types, and text editing. You can easily switch between orientations by choosing the opposite split type in the **View ▶ Layout ▶** sub-menu, or Option-clicking the split icon in the Header Bar (section 14.1).

When the editor has been split, a header bar and footer bar will be created for each split. This means that each side has its own history queue, view modes, zoom settings, outliner columns and so on. Everything that you can do to a single editor window can be done to a split window, and those changes will be remembered for *that side* of the split. An example of this would be view modes. With a single editor interface, if you choose to view containers using the Corkboard model, every time you click on a container or select more than one item, a Corkboard will be presented to you. However, when splits are engaged, each side has its own view settings. This means the left side (in a vertical orientation) can be set to Outliner, and the right side set to Scrivenings.

You can adjust the comparative size of the splits by dragging the strip along the middle that separates them. This can be done freely, though there are limits to how small a split can be made. MAC: To reset the splits to be of equal width or height, double-click the splitter bar with your mouse.

> **Quickly switching between editors**: It is easy to jump between splits without using the mouse, by using **^⌥⌘-E** and **^⌥⌘-R**. When the split orientation is vertical, E will access the left editor and R the right; when the split orientation is horizontal, E will access the bottom editor and R the top. Additionally the keystroke that cycles between the Binder and the Editor, **^Tab**, will include both splits in the rotation.

14.4.2 Controlling the Opposing Split

There are a few commands that you can use to impact the editor your are not currently working in, reducing the need to flip back and forth between them. These are located in the **View ▶ Editor ▶ Other Editor ▶** sub-menu, and have shortcuts for handy usage.

Remote scrolling To scroll the other text editor up and down, use **^⌥⌘-UpArrow** and **^⌥⌘-DownArrow**. This command does not work when the other editor is viewing Corkboard or Outliner views, or media that would not otherwise respond to scrolling.

Note that this form of scrolling is based on the scrollbar, not actual text position, so the position of the scroll point will not be saved when used in conjunction with the history button, like cursor position is. The exception to this are PDF files, which are always saved by scroll-point, not cursor position.

Remote history access Just as you can quickly flip through the history with ⌘-(and ⌘-), you can cause the other editor the jump back and forth in its own history queue with ⌥⌘-(and ⌥⌘-). A nice trick here, when working with a sequence of reference documents, is to "pre-load" them into your reference split by clicking on each one sequentially. Now they are loaded into the history queue and easily accessible while you write in the other split.

Controlling media While typing in one split, you can start and stop QuickTime movies and audio tracks with ⌘-Return, making this setup extremely useful for transcription. If you have Rewind on Pause enabled (section 14.3), this shortcut will also rewind the piece by a set amount (3 seconds by default), making it easy to catch up.

14.4.3 Managing Split Views

If you wish to swap the actual position of the material in the editors, so that the content on the left now appears on the right, for example, use the View ▸ Layout ▸ Swap Editors menu command.

To synchronise the contents of both splits, click the document icon in the header bar for the side you wish to use, and select the Match Split Document(s) command. This will duplicate your current view selection in the other editor. So if you are in a Scrivenings session, that session will be duplicated; if you have a Multiple Selection view, that selection will be duplicated in the bottom view—and the current view mode in the active split will be used instead of the active default for that split (and will thus set the default for that split to the acquired view model for purposes of future navigation).

Chapter 15

Writing and Editing

Scrivener uses the standard OS X text editor, and therefore all of the features of OS X's rich text editing system (which are showcased in Apple's TextEdit application) are available[1]. Scrivener also provides some extra word processing features. Most of these features are available via the Format (section A.7) and Edit (section A.3) menus, so it is recommended that you familiarise yourself with these.

> **The text engine, and Mac OS X**: Please note that because Scrivener uses the OS X text system, its behaviour in this regard is defined by Apple. The way double- and triple-clicking on text works, the occasional awkwardness of features such as tables and bulleted lists, font changes and so forth, are all governed by the OS X text system, which is programmed by Apple and is out of our hands. The good news is this also means you get access to many sophisticated tools which would otherwise take years to program.

15.1 Rich Text Editing Philosophy

Scrivener supports a rich text editing environment, which means that it is loosely "what you see is what you get". Unlike word processors or desktop layout applications, however, the precise formatting that you use when writing in Scrivener may in fact look nothing at all like the final product. The compiler will be covered in greater detail in a later section (chapter 24), but suffice to say that you can work in one font, say the default Cochin, but publish in an industry standard font like Times New Roman, without having to change your source text. What this means for you is that certain aspects of your editing can be conveyed through the compiler, like a range of italicised text, while the base font itself can be changed beneath that. Scrivener's editor can thus be used *like* a typical word processor, but with the knowledge that rich text formatting can be altered later on to produce different results.

[1] The only exception to this is that in Scrivener, you are not allowed to paste QuickTime files into the text. This is because of a bug in OS X that can cause crashes or strange behaviour when a QuickTime file is contained inside a single piece of text that is being viewed in two panes, such as Scrivener's split view.

Scrivener is not intended to be a full-blown word processor, but rather a word generating environment. It is not a layout tool, but a tool for cutting the text that will become your book. Many authors who have stricter formatting requirements, such as scriptwriters, will start their projects in Scrivener, and end in a word processor or desktop publishing, or some other specialised application. Someone writing a novel or short story could very well do the whole thing in Scrivener, including the final print for submission and export to Word format for sending an electronic copy.

You might be wondering if Scrivener offers a plain-text editing environment as well. There is no plain text option, as Scrivener is a writer's tool rather than a strict text editor, and many of the tools that are made available to you for writing rely upon rich text to function, such as highlights, marking revisions, and annotating. Even if you do not require formatting, you will probably find the formatting tools *useful* in the writing process, especially if you are compiling to plain-text, where they will all be safely stripped out and invisible to the reader.

> **Emulating plain-text**: If you find working in a plain-text environment to be more productive, you may wish to set the default editor font to one that does not support rich text features, such as Courier, and change the ruler settings so that there are no indents or paragraph spacing. This set-up will emulate a plain-text environment whilst still allowing features such as annotations and highlighting.

When using MultiMarkdown to format your work, you will find that some formatting, such as inline images, footnotes, and so forth will be converted to MMD markings during compile. Most formatting, however, will be stripped and thus can be useful in an annotation setting.

15.2 Editing Basics

15.2.1 Caret Movement and Selection

The following list shows the shortcuts for changing the current selection. To simply instead move the current insertion caret position, omit the shift key where applicable.

- ⇧**ArrowKeys**: Extend current selection in the direction of the arrow key that is used.
- ⇧⌥**LeftArrow or RightArrow**: Extend the selection by word.
- ⇧⌥**UpArrow or DownArrow**: Extend the selection by paragraphs.
- ⇧⌘**–UpArrow**: Select from the caret position to the top of the editor.

- ⇧⌘–DownArrow: Select from the caret position to the bottom of the editor.
- ⇧⌘–LeftArrow or RightArrow: Select from caret position to beginning or end of line respectively. Note the definition of line is not a paragraph, but is instead dependent upon the current soft wrap. To select to the beginning or end of the paragraph, use the above paragraph selection shortcut.
- ⇧MouseDrag: Extend the current selection using the mouse.
- Double-click: Select word. Can be used in conjunction with dragging to select a range by word.
- Triple-click: Select paragraph. Can be used in conjunction with dragging to select a range by paragraph.

An advanced method of selection can allow you to select more than one location at once. By holding down the Command key and selecting using the mouse, you can select several areas of non-consecutive text. The Option key can be used to select rectangular portions of text, which is mainly useful for trimming unwanted characters off of the beginnings of several lines.

- **Jump to end of current paragraph**: ^E. This is equivalent to ⌥DownArrow, but is significantly easier to input as it uses the typing keys.
- **Jump to beginning of current paragraph**: ^A. Again, equivalent to ⌥UpArrow but requires less hand movement while typing.

15.2.2 Deletion

Simple character deletion can be performed with the delete key to delete the character prior to the caret position. On extended keyboards, an additional DEL key (which deletes the character *after* the caret position) is provided in the central auxiliary cursor control area, between the alphabetic keys and numeric keypad. On many compact and laptop keyboards, you can activate forward-deletion with the Fn-delete combination.

- **Delete to beginning of current word**: ⌥delete
- **Delete to end of current word**: ⌥Fn-delete or ⌥DEL
- **Alternate forward character delete**: ^D. This is the same as DEL or Fn-delete, but you might find it easier to use since it doesn't require your hands to be substantially moved from the home row.

- **Delete to beginning of soft wrapped line:** ⌘–delete
- **Delete to end of paragraph:** ^K. Do note this deletes all the way to the end of the paragraph, not the current soft wrap line.
- **Transpose letters around caret:** ^T

15.2.3 Spell Checking

Spell checking in Scrivener can be accomplished either as you type, or after you are done writing a section. Depending on your preferred way of working, you'll want to set how this works in the Corrections preferences tab (section B.9), where you will find other options such as which language to use, and whether or not Scrivener should try and fix common typos for you as you write. You may also use **Edit ▸ Spelling and Grammar ▸ Check Spelling While Typing** (⌘–\) to toggle instant spell checking on and off for the project.

By default, Scrivener will highlight misspelled words as you write. If you would prefer to not be alerted to misspellings while writing, you can defer the spell check process for later. When you are ready to spell check a document, right-click anywhere in the document, and select **Spelling and Grammar ▸ Show Spelling and Grammar** from that menu. A familiar floating window will let you step through the document error by error, fixing mistakes and adding unique words to your personal dictionary as you go. As with most tools of this nature, if you type anything into the top field and click "Change" you can add a correction that is not shown in the suggestion list.

The **Ignore** button will temporarily ignore the misspelled word, while **Learn** will add the word to Scrivener's permanent ignore list. The **Guess** button can come in handy if none of the suggestions are correct. You can change the word itself in the top field, or select one of the suggestions and click this button in an attempt to narrow the list down to the correct word.

Grammar checking can also be enabled here, and you can look up definitions for words by clicking on them in the suggestion list and clicking the **Define** button. This will open the standard Dictionary tool, with the word pre-selected.

Spell Check Contextual Menu

With active spell check while typing enabled, when you misspell a word while writing, it will be underscored with a wavy red line. Whenever you see a word with this marking, you can right-click on it and the contextual menu will contain best-guess suggestions for which word you were aiming for.

The contextual menu contains a few convenience commands:

Ignore Spelling Temporarily ignores the word for the current session, in all documents within the project. Note that in some cases you may need to "refresh" the spelling engine by viewing another text document and then returning, as ignoring by right-click will often only impact that one instance you clicked upon.

Learn Spelling Permanently adds the selected word to your personal dictionary. Since Scrivener makes use of the global Mac spell check engine, this will add the word to your entire Mac, meaning it will no longer appear misspelled in Scapple, Mail, Safari and other native Mac software.

Unlearn Spelling This option will only appear when right-clicking on a word that you have previously "learned". This will remove the word from your personal dictionary.

15.3 Editing with Scrivener

Beyond the basics of text editing, which are similar to many applications, Scrivener provides further tools, specifically designed for writers, in its editor interface. The rest of this chapter will focus on these tools in a comprehensive fashion, where you can glance through the list of topics covered and pick which items you feel are best suited to your writing style and the task in hand. You could read this section from start to finish, but it is meant to be a collection of individual nuggets that you can learn independently, as you gradually build up your knowledge of the application.

15.3.1 Scaling Text

The text of the editor can be scaled up and down, without affecting the font size, via the ability to zoom the scale of the text. Each split can have its own zoom setting; additionally the composition mode can have its own independent zoom setting as well. All of these settings are preserved between sessions and saved into the project file itself.

In the standard text editor interface, zoom can be set using the zoom tool in the footer bar (section 14.2). In composition mode, this same tool is provided within The Control Strip (section 17.2).

You can also use the menu commands:

1. **View ▸ Zoom ▸ Zoom In** (⇧⌘–.) to increase the zoom factor of the text.
2. **View ▸ Zoom ▸ Zoom Out** (⇧⌘–,) to decrease.

If you prefer to always work from a certain zoom setting, the default zoom level for both standard editors and the composition mode editor can be adjusted in the Editor preference tab (section B.7), and the Compose preference tab (section B.5), respectively. Note

that this setting will impact all *new* projects, or recently upgraded projects, not projects that have already been created, as each project can save its own individual settings.

Additionally, Inspector footnotes and comments can have their text dynamically scaled, either to match the current text editor, or at a fixed percentage value (subsection 18.2.4).

15.3.2 Contextual Menu for the Text Editor

The text view's contextual menu contains many standard commands such as Cut, Copy, Paste, Spelling and so forth, along with a number of commands specific to Scrivener. Several common tools, such as Spotlight searching, dictionary access will be located in the "Writing Tools" sub-menu. If a selection has not already been made, right-clicking will select whatever word (or words connected by a hyperlink) is under the mouse pointer.

The Scrivener-specific commands (the appearance of which will depend on the selection) are listed below:

Text Editing Contextual Menu

Remove Link When the active selection contains one or more links, you can strip the links out with this command.

Look Up 'word' [10.7+ (Lion) Only] When right-clicking on a word, or a selected phrase, the "Look Up" menu item will be inserted at the top. This will open the Dictionary and Thesaurus pop-over that can otherwise be accessed with ^⌘-D when hovering the mouse over a word.

Select Annotation/Footnote When right-clicking within an inline annotation or footnote, this option will appear, making it easy to select the entire contiguous range of text that has been marked as notation. When two inline annotations of different colour fields are located adjacent to one another, this command will only select up to the edge of the current colour right-clicked upon.

Split at Selection (⌘-K) Splits the current document into two documents at the selection point (the current blinking cursor point, or the initiating edge of the selection, which will be on the left by default, or on the right when using right-to-left languages).

Split with Selection as Title (⌥⌘-K) Splits the current document into two documents using the current selected text as the title for the newly created document. The selection will remain after splitting, making it easy to remove the redundant title text if necessary, or style it like a header. For more details on splitting documents, read Managing Documents with Split and Merge (subsection 15.3.3).

Append Selection to Document Provides a menu of all documents in the binder. Selecting a document from this menu will cause the selected text in the editor to be appended to the document selected from the menu. Read Gathering Material (chapter 11) for more tips on moving text around and organising information in the Binder.

Set Selected Text as Title (⇧⌥⌘–T) Sets the title of the current document to the text selected in the editor.

Add Selection to Auto-Complete List Adds the selected text to the project's auto-complete list, which can be maintained with the **Project ▸ Auto-Complete List...** menu command. Read more about this feature in Auto-Completion (section 15.7).

Link to Document Allows you to create a link to another document in the project within the text. Read more about linking items together in Linking Documents Together (section 9.5).

Text Color Provides the text colour menu, from which you can select from the built-in colours, or those custom colours you have saved into your colour palette.

Highlight Color Provides the highlight menu, allowing you to select a highlight colour for the selected text, or to clear it. Read more about this and the above feature in Text Colour and Highlights (section 18.4).

Spelling and Grammar Quick access to spell check tools.

Apply Formatting Preset Apply one of your saved formatting presets (subsection 15.4.3) to the selected text.

Font Provides basic character attributes, such as bold and italic, as well as access to the main font chooser.

Speech Quick access to the **Edit ▸ Speech ▸** sub-menu.

Writing Tools ▸ Quick links to a few handy web searches and the system dictionary and thesaurus application. The current word, or selected text, will be sent to the search engine or reference site of your choice.

Hyperlink Selections

When the selection is entirely or partially within a hyperlink (not an internal Scrivener Link), then additional link management options will be provided for at the top of the contextual menu:

Open Link The same action as left-clicking on the link with the mouse. This method is here primarily for accessibility purposes.

Copy Link Copies the underlying URL of the link, as in many web browsers, so that it can be pasted bare or used in other contexts such as the creation of new hyperlinks, or to import the web page into Scrivener using File ► Import ► Web Page....

Edit Link Brings up a window for editing the underlying URL of the link. This is the same as using Edit ► Edit Link....

Images Contextual Menu

These options appear when right-clicking on an image that has been placed within the text (section 15.5).

Save As Picture... Available if you have clicked on an inline image. Allows you to save the image to disk. You can also drag images into the Binder.

Scale Image... Available if you have clicked on an inline image. Brings up the image scaling panel.

Linked inline images (subsection 15.5.3) provide a few extra convenience commands for working with the file on the disk:

Reveal in Finder Opens a new window in your computer's file manager, with the linked image highlighted in the window.

Open in External Editor The file will be opened directly off of the disk using the system's default image editor. This will edit the *original file*, since it is merely linked into Scrivener.

Scrivener Link Selections

Open Document Link In ► This command only appears when right-clicking on a single Scrivener Link, where only that link is selected and no extra text around it. This is most easily accomplished by right-clicking on the link itself (the selection will expand to include the full link range automatically). The contextual sub-menu gives you a few extra options for where the link should be opened, letting you override your application default as set in the Navigation WIN:optionsMAC:preference tab (subsection B.6.7).

Update Links to Use Target Titles Only appears when one or more Scrivener Links are contained anywhere within the current selection. This command updates the link text to match the binder titles of the items they respectively link to. Read more about the feature here (subsection 9.5.4).

Tables Contextual Menu

When right-clicking within a table in the editor, additional options will be provided in the contextual menu:

Table... Accesses the table palette, which provides formatting, cell dimension, and nesting, and cell split and merge features. Inserts a table if one does not already exist.

Add Row Above/Below Will insert a new table row of empty cells above or below the row in which you right-clicked.

Add Column Before/After Inserts a new table column of empty cells to the left or right of the column in which you right-clicked.

Delete Row/Column Will delete the entire row or column in which the cell you right-clicked upon is located.

Sort Rows Ascending/Descending Sorts all rows in the table based on the alphanumeric value of the text in the row that the selection is currently in.

Borders This sub-menu provides some handy border tools.

Distribute Rows/Columns Evenly Adjusts the height or width of all rows or columns so that every cell has the same dimensions.

Remove Table Strips the table code out from the currently active table, leaving all text behind as linearised.

Lists Contextual Menu

Only one extra option over the base text system has been added for list management:

Re-number List In rare cases, especially when pasting lists from other word processors like Word, list numbering will sometimes not register properly. Use this command to attempt to repair these lists.

15.3.3 Managing Documents with Split and Merge

The ability to expand and contract the outline in a fashion which best suits how your work is important, so there are two tools which make it easy to fashion your outline into as broad or detailed a map as you require.

Before discussing the specific tools which let you cut apart documents or sew them back together, it would be useful to also discuss a few methods that you can use to mitigate the regular need for them.

When you wish to merge two or more documents together, it is good to ask whether or not this action is necessary. With most applications, and thus the habits we have cultivated from using them, the answer to whether or not one should merge is, yes. However Scrivener offers two powerful answers for the merging problem. First, for visualising smaller pieces of a document as a single document, but only temporarily, you can select any container in the binder, or even just a scattering of items, and choose to view and edit them much as though they were a single document (section 15.9). Second, the compilation (chapter 24) system enables you to publish your final work as a single document, no matter how many pieces it may be divided into within Scrivener.

For splitting long documents, in some cases you may be able to utilise use import tools which do most, if not all, of the splitting for you. **File ► Import ► Import and Split...**, will let you supply a separator, or a string of characters, that mark natural sections within the original file. Whenever Scrivener encounters this separator, it will automatically start putting the remaining content into a new binder item, continuing to split into new binder items until it runs out of material. The second tool, special for MultiMarkdown users, is **File ► Import ► MultiMarkdown File...**. This tool will examine the incoming file for its header structure, and not only split files between headers, but title them according to those headers, and even reconstruct the hierarchy of the original document. Finally, if you've done your original outlining in a dedicated outliner program, it will likely support an OPML export, which you can drag into Scrivener's binder. It will recreate the outline structure, placing any original outline notes into the text area for each outline header.

For most other cases, read on.

Splitting The Document

Often, when importing material that you have composed in other applications that do not support an easy to use built-in outline, your text will arrive in Scrivener in one large chunk. Where you choose to break this long document apart is up to you, and the split tools make it easy to do so.

To split in a document into two pieces, you will first need to place the caret at the precise point in the document where you wish the split to occur. If you select a range of text, the *start* of the selection will be considered the caret point, for purposes of splitting.

Once the selected location has been chosen you can use one of two methods to split the document:

Documents ▶ Split ▶ at Selection (⌘–K) This will create the new document directly below the current document in the outline. It will contain everything that followed the caret position, and this material will be removed from the original document. It can be useful to think of splitting in terms of "above" and "below" the selection. Everything above the caret (and also to the left if in the middle of a line) will remain in the original document, while everything to the right and below of the caret will be moved to the new document.

After splitting, you will be given the option to name the new document in the Binder.

Documents ▶ Split ▶ with Selection as Title (⌥⌘–K) This alternative method will only appear when a range of text has been selected in the editor. This method works identically as "at Selection" does, only the selected text will automatically become the title of the new document. This text will remain selected after you split, making it easy to remove it, or style it as a header.

When splitting documents, most meta-data will be carried over to the new document. The only exceptions to this will be any snapshots and the synopsis. Snapshots will remain attached to the original document (everything falling before the split point), and can thus be useful in seeing what a document looked like before it was split up. It is thus a good idea to set up any common meta-data like keywords and labels before you start splitting up a document, to avoid having to make numerous identical changes later on.

It is not possible to undo a split action, but you can use the Merge feature to effectively undo any unwanted splits.

> **Pro Tip**: You might also find the ability to search by formatting (subsection 21.1.6) to be of use as well, as often the places you will want to split the document will coincide with headers. While the Find by Formatting panel is open, you can use the **Split with Selection as Title** function without closing it, and then quickly go to the next search result within the portion that has been split off. Or, if you wish to use keyboard shortcuts, this means you can alternate between using ⌥⌘–K to split the found text, and ^⌥⌘–G to skip to the next search result, even with the Find by Formatting box closed.

Merging Documents Together

In opposition to splitting, the ability to select two or more documents and merge them together into a single document is also made easy with Scrivener. Unlike the split func-

tion, merging is a super-document level action, and thus requires a selection to be made in a corkboard, outliner, binder view, or from within a collection; you cannot merge from within a text file.

Documents do not have to be in consecutive order, they can be picked from throughout the project. When selecting non-linear items, here are some tips to determine ordering:

- If the view you are picking documents from is based on the outline order, such the binder, outliner, or corkboard (for this purpose, a freeform corkboard is still considered to adhere to Outline Order. If you wish to impose a new order based on the freeform spatial placement, make sure to use the **Commit Order**button (section 12.1.4), first, and then merge), then the merged document will retain the original outline order.
- When using a Collection to select items, the *collection order* will be used to established the structure of the merged document.

When it comes to merging, Scrivener will attempt to retain as much meta-data as is logically possible. The synopses, notes, keywords, references, and list of snapshots will be combined together, much in the same fashion that the main text will be. Meta-data which cannot be combined (such as Title, Label, any custom meta-data, compile option flags, and so forth) will use the top-most document as a reference point. For example:

- Document A
  - Synopsis: “This is the first document”
  - Keyword: “Apples”
  - Status: “First Draft”
- Document B
  - Synopsis: “This is the second document”
  - Keyword: “Carrots”
  - Status: “To Do”
  - Notes: “Mix these two together in a blender for a great tasting, healthy beverage.”

Both Document A and Document B are selected in the Binder, and **Documents ▶ Merge** is invoked (⇧⌘–M), producing the following single document (with the bold items indicating material used from Document B):

- Document A
  - Synopsis: "This is the first document

    This is the second document"
  - Keywords: "Apples", "**Carrots**"
  - Status: "First Draft"
  - Notes: "**Mix these two together in a blender for a great tasting, healthy beverage.**"

When merging, the main text will be combined according to the preferences you have set in the General preferences tab (section B.2), under the Separators section. Merged documents can be combined with a single line, a full empty line (useful in situations where plain-text is in use and paragraphs need to have an empty line between them), or a custom separator, which allows you to type in any sequence of characters. Note you can add carriage returns and tabs to this field by holding down the Option key and typing in the special character.

15.3.4 Typewriter Scrolling

You can turn "typewriter scrolling[2]" on for the main editors, QuickReference panes, and for the composition mode editor independently via the Format ▸ Options ▸ sub-menu (^⌘–T). When it is switched on, just as with a typewriter, the line of text you are typing will remain centred vertically in the editor, once your text is long enough to reach the middle of the screen. This means that your eyes do not have to be glued to the bottom of the screen all the time when typing in a long piece of text.

When Typewriter Scrolling is enabled, your scrollbar will act as though there is more content than the document actually has. This is because we add extra spaces below the last line in the document in order to keep it in the middle. These spaces do not actually exist in the file, and will vanish when the feature is disabled.

It is possible to adjust where the focus point will be. If you'd prefer for editing to happen in a different fraction of the screen, visit the Editor preference tab (section B.7), and set the "Typewriter scroll line" to the top quarter, third, middle (default), bottom third, or quarter of the screen.

The Typewriter Scrolling feature is not available in Page View, given how this feature lays out the pages in a simulacrum of the printed page, which will of course not contain any of the dummy lines added to keep the entry position in the middle of the screen.

[2] Typewriter scrolling was an innovation of the The Soulmen's *Ulysses*.

Figure 15.1: Example Ruler Settings

15.4 Formatting Tools

15.4.1 The Ruler

The text editor in Scrivener uses a ruler for setting indents and tab stops, such as many word processor applications provide. It can be shown or hidden via **Format ▸ Show(Hide) Ruler** or ⌘–R. This will present a simple indenting and tab stop interface (Figure 15.1).

The following elements can be found in the ruler:

Tab Stops Left stops can be created by clicking in the numbered area and dragging up, or by right-clicking in the ruler and choosing a type. Once placed, they can be moved via click-and-drag, and the numerical value of its position will be printed above the mouse pointer as you drag. These will be depicted by type using icons. The four tab stop types available are:

- Left: This is the standard type. Text will be left-aligned, with the first character indented to the position of the tab stop.
- Center: Text will be centre-aligned, using the position of the tab stop as an anchor point; this can be anywhere on the line.
- Right: Will right-align text with the right-indent set to the point of the tab stop.
- Decimal: Most often used for aligning rows of numbers, so that the system decimal character is lined up vertically with everything before the decimal being right-aligned, and any fractional values left-aligned.

To remove a tab stop, simply click and drag it down or up out of the ruler until the icon disappears.

Indents There are three indenting controls, not to be confused with margin controls (which Scrivener's ruler does not address). Indenting is the action of offsetting text

a defined distance from the margin. A left indent pushes the text boundary toward the right, away from the left margin. A right indent pushes the text boundary toward the left, away from the right margin.

In addition to the two primary indent marker is the first-line indent marker. This will *only* indent the first line of a paragraph. Using this control, it is possible to set the first line to indent by half an inch, while the rest of the paragraph is set to zero (or directly adjacent to the margin).

Scrivener's default formatting settings demonstrate the best way to use indentation, if you need to use it at all.

Hanging Indents These are produced using the same tools as ordinary first-line indents, only to produce a hanging indent, you will need to inverse which marker comes first. If the first-line indent marker is to the left of the left-indent marker, then the body of the paragraph will be pushing inward, while leaving the first line "hanging" over the blank space left below it.

Right Indents These are rarely used, as they will often restrict flexibility down the road, depending on your ultimate page size, and it can be confusing to calculate where they should be set to. For instance, if you prefer to proof with a more generous page margin of 1.75" but need a 1" margin for submission, an appropriate level of right-indentation away from the margin would differ. However there are cases where it is necessary, of course, and in such cases you will need to restrict yourself to using a more limited set of paper and margin sizes.

Sometimes, when importing material from word processors you may get text which is already indented artificially, causing it to wrap before it gets to the edge of Scrivener's editor window. You can either pull the right-indent marker all the way to the right to reset the value to off, or use the **Documents ► Convert ► Formatting to Default Text Style** menu command to clean up large amounts of text at once.

Ruler Conversions: When working with units in Scrivener, keep in mind that its ruler starts at margin zero instead of paper zero. Since Scrivener is, by and large, not "aware" of paper settings and page layout, it counts its ruler settings from the beginning of the text on the left end of the page, not the beginning of the paper itself. This is in contrast to many word processors, which start measuring at the paper left edge, and show the print margin buffer in the display of the page.

Consequently, to convert most standard measurements to useful values here, you will need to factor in the standard amount of print margin by adding in the margin to the value you see in the ruler. For example, if you are using 1" margins and you

Figure 15.2: Format Bar: displayed in sections

need a right-indent of 1", the right-indent should be set at 5.5" for US Letter, not the standard 6.5". Optimally, your formatting can be deferred and set in the compiler, where formatting and paper dimensions can be operated upon in concert.

Using tab stops while writing is as simple as pressing the tab key to advance to the next available stop. If there are no more tab stops available on that line, the system will wrap around to the next line at the first tab stop, but no newline will be added. Tabs allow you to enter tabular information into a single line, without using tables. To remove a tab from the line, simply delete it as you would any other character. Tabs can be viewed as symbols with **Format ▶ Options ▶ Show (Hide) Invisibles**.

Ruler settings are adjusted per-paragraph. If you wish to set tab stops or indents for many paragraphs at once, you will need to select them prior to using the ruler. If for some reason you need to apply ruler settings from one portion of text to another, it is possible to copy paragraph settings by using **Format ▶ Text ▶ Copy Ruler** and **▶ Paste Ruler** menu commands.

15.4.2 The Format Bar

The Format Bar provides quick access to common formatting features. If you are looking for tab and margin controls, you can still toggle that per text view with **⌘–R**. The visibility of the Format Bar is toggled with **⇧⌘–R**.

The first section (Figure 15.2) of controls handle font face, size, and variant. All of these tools act immediately on the currently selected text. If no selection is given, then they will alter how you type from the current caret position onward.

- The first drop-down contains formatting presets (subsection 15.4.3); handy ways to apply many formatting preferences at once.
- The second menu lets you select from all of the available font families installed on your computer.
- The third provides the variants available for that family. Note that not all fonts come with a full complement of variants. If for some reason you cannot change

a selection of text to bold or italic, check this menu to make sure that the current font supports these variants.

- The fourth menu provides a quick list of common font sizes. If you need a custom size that is not available on this list, use the **Font Panel...** item at the bottom to access the standard OS X font palette (subsection 15.4.4).

The aforementioned Presets menu hooks into Scrivener's formatting preset system. As is the case with Apple's default system, it is important to note that these are format and ruler "brushes", not word processor stylesheets. They apply saved formats to the selected text or the caret position, but once these have been applied they will not automatically update later on if you change the preset.

The next three buttons (section two (Figure 15.2)) provide toggling shortcuts to the most common text formatting tools: bold, italic, and underscore, respectively. When the caret is in a range of text that matches the format style, the appropriate buttons will appear activated.

Directly to the right of these, control for paragraph-level text alignment exist in button form. Paragraph-level alignment comes in four flavours:

- Left Align (default)
- Centre Align
- Right Align
- Full Justification

In all cases, these adjust the entire paragraph. If you need multiple alignments on a single line, you will need to add special alignment tab-stops to the ruler (⌘–R).

In the third section the left colour box will alter the foreground text colour. Click the button once to set the selected text to the depicted colour (black by default), or right-click to access a pop-over with convenient colour selection utilities. You can opt to strip out all colour code from here with the top-left square that has a red slash drawn through it, as well as access the standard colour palette and any custom colours you've added to it. For more information on using the colour tools, see Text Colour and Highlights (section 18.4).

The second colour selection drop-down works in an identical fashion to the first, but it sets the background highlight colour for the selected range. Click once to toggle the current highlight (or use ⇧⌘–H) for the selected text, or use right-click to access further options as described above.

The final, fourth section of the Format Bar accesses paragraph & line spacing, and lists.

The spacing drop-down provides some quick presets for global line spacing, but for more complex multi-type spacing preferences, as well as paragraph spacing (above or below), select the **Other...** item at the bottom of this menu.

The list tool provides most common Western list formats. For custom and international formats, select the **Other...** menu item. Lists can be enumerated or bulleted in a variety of methods. They will transform the current paragraph (even if empty) into the first list item, and will continue generating new list lines until you enter a blank line. While entering lists, it is possible to indent or outdent lists with **Tab** and ⇧**Tab**.

15.4.3 Formatting Presets (Styles)

For occasions when you need to apply a certain style of formatting repetitively throughout your work, a tool that can store formatting settings and apply them later to other ranges of text will make writing in Scrivener more convenient.

> **Presets are not Stylesheets**: If you are familiar with using stylesheets from traditional word processors, you might be looking for something similar in Formatting Presets. While presets do make the propagation of formatting easier, they do not assign the target text with any kind of named range, like stylesheets do, so that they can be universally updated centrally by the stylesheet. Once applied, the text merely ends up looking as the preset ordains it to, with no further connection to that preset. It is thus a good idea to make sure you have your formats locked down as close as possible, if you intend to use this feature to generate output-ready content.

A few Presets have been provided to you for convenience, or as starting points for your own preferred look and feel. Note that Presets are not in any way tied to the rest of the application or project formatting rules. For instance, if you change the application default to use paragraph spacing instead of first-line indentation, the Body preset will not change and will still apply an indent to the target text. This is by design, as "Body" is nothing special and can even be deleted or completely redefined.

There are two basic types of presets available:

1. Paragraph style: this will save all of your ruler settings, as well as alignment and line-spacing attributes, and anything else found in the **Format ► Text ►** sub-menu.
2. Character attributes: this will save any format level adjustments, like bold, underscore, kerning, or anything else found in the **Format ► Font ►** sub-menu.

These two types can be used exclusively, or combined into a single Preset, controlling all aspects of the text appearance, and both can optionally also apply font size and font

family information. The Preset menus will display small icons in blue, next to each entry, indicating whether the style will impact one or both of these types when applied to your text.

Applying Presets

To apply a preset, first select the text you wish to apply the preset to, then use one of the following methods:

- Use the Formatting Presets button on the Format Toolbar to select a preset from the drop-down menu.
- Use the **Format ► Formatting ► Apply Preset ►** sub-menu to select the preset, or use the "Apply Formatting Preset" sub-menu from the right-click menu.

The types of formatting applied to the text you have selected will depend upon the type of preset. For character attribute presets, you must select the text you wish to apply the preset to, just as you would when using bold or italics. For paragraph style presets, the cursor can be anywhere in the target paragraph and no explicit selection need be made. Of course, if you wish to change many paragraphs at once you can use standard selection to do so. Whatever paragraphs the selection *falls into* will be impacted, even if they are not fully selected.

Presets which contain *both* character and paragraph attributes will apply all stylings to the entire paragraph, regardless of selection. If you wish to only apply font styles to a portion of your paragraph, then you will need to create a character attribute preset to do so. For example, if you saved the bold attribute into the paragraph and character preset, the entire paragraph will be emboldened, regardless of the selection you've made.

Creating New Presets

To create a new preset, you will need to create exemplar text in your editor, containing the styling you wish to preserve for future use. Place the cursor in the text where that format appears. It is often best to not *select* any text for this procedure, as doing such will increase ambiguity where multiple styles and attributes are employed within the selection.

Invoke **Format ► Formatting ► New Preset From Selection**. You will be presented with a sheet that will let you name the new style, choose whether it will impact character attributes, paragraph styles, or both, and optionally if font settings should be stored as well. If you do not want the preset to change the underlying font family or size, then make sure these checkboxes are disabled. Once a name has been added to the sheet, you

will be able to save the preset for future use. It will appear in the Format Bar menu, as well as in the application menus.

Modifying a Preset

To modify an existing preset:

1. Apply the preset to some sample text and then modify the styling of this text in the editor using the standard formatting tools.
2. With the cursor still located within the sample text, use the **Format ▶ Formatting ▶ Redefine Preset from Selection ▶** sub-menu to select the preset you wish to modify.

You will be presented with the same sheet you used to create the preset, only the information you saved into it will be pre-populated into the sheet. In most cases you will not need to change anything here, and can just submit the form to apply the formatting changes you've made. If you do wish to change the parameters of how much the style operates, you can make these changes to the sheet, or even change the name of it.

If you wish to create a derivative, instead of replacing an existing preset, use the preset to style your text, modify it, and then use the **Format ▶ Formatting ▶ New Preset From Selection** menu command to create the new version.

Deleting a Preset

Deleting a formatting preset can be done at any time by using the **Format ▶ Formatting ▶ Delete Preset ▶** sub-menu. This command cannot be undone, but if you do accidentally delete an important preset, you can re-create it by finding an example in your text, and creating a new one from scratch.

15.4.4 Font Palette

The font palette is a standard tool provided by Apple, and can be toggled with **Format ▶ Font ▶ Show Fonts** and **▶ Hide Fonts**. You can also use **⌘–T** to toggle this view at any time. This palette provides full access to the font typesetting engine, including access to many OpenType features and custom font effects. For basic font changes, the Format Bar (subsection 15.4.2) will suffice.

The precise appearance of the palette will change depending how large you make it. As the size of the palette increases, more options will be made available to you. Since this palette is provided by Apple, please refer to the help files provided by TextEdit, for details in using it.

To use the font palette to change the appearance of your text, you will need to first select the text you wish to change, and then call up the palette. Any changes you make will be immediately reflected in the editor.

15.4.5 Resetting Formatting

Oftentimes, after you've gathered material from the Web, or imported documents that you've written in another word processor, the formatting of the imported material will not match the default font in Scrivener for new documents. While you can change this default at any time with the Formatting preference tab (section B.8), this will not impact documents you've already created or imported, as Scrivener has no way of knowing if that is what you really want to do. Often it might be okay to leave the file as non-standard, especially if it is research material that you never intend to include directly in the draft or even edit.

If you do want to retrofit these documents to your current defaults, you can do so with the **Documents ► Convert ► Formatting to Default Text Style** menu command. Since this command works at the document level, you can select as many cards, outliner rows, or binder sidebar items as you please and convert them all in one go.

After clicking the menu command, a window will pop up asking how much formatting you wish to apply to the selected document(s). Most of these options work in a *negative* fashion. By example: if you select "Preserve alignment" and the document is left-aligned, even if your preferences are for full justification, it will not be applied because you have elected to preserve the original alignment. The only exception is the "Convert font only" option, which acts as a handy way of preserving all paragraph level formatting, without individually checking off their options.

Once you've used this window to set up how deeply this command will impact a document, you can waive it in the future by holding down the Option key when selecting the menu item.

Please note that because this command impacts broad changes in potentially many dozens or more of documents at once, there is no undo. The procedure only impacts formatting, so it is safe to use, but if you are unsure of whether or not the result will favourable, either Snapshot (section 15.6) the documents first, or perform the conversion one-by-one and proof the results. Once you are confident you have the right settings engaged, you can proceed at a more rapid pace.

Some simple tips for preserving common formatting based ranges:

- **Preserving Block Quotes**: Block quotes are usually indicated entirely by some form of increased left, and sometimes right as well, indent. Preserving tabs and

indents will protect most block quotes from being reduced to normal text by this tool.

- **Preserving List Spacing**: Some styles call for lists to be single-spaced even if the text around them has a greater line-height or inter-paragraph spacing. Using "Preserve line spacing" will protect these lists at the expensive of not applying your preferred line spacing to standard paragraphs.

- **Preserving Titles**: If you've got a document with a lot of headers already formatted into it, you could try preserving font size, style and line spacing, but this will in many cases be the three things you *want* to apply. Titles are going to be harder to manage automatically, and may require manual effort to keep their formatting as intended. Another thing you can consider doing is switching to using Scrivener's philosophy of using the compile feature to generate titles from your binder names, and leave worrying about special formatting for headers to the computer.

Ranges of text that have been blocked out with Preserve Formatting (subsection 15.4.6) will never be altered by this tool, much in the same way that the compiler will leave them alone by default.

One last tip is to use "pre-emptive" format conversion. If you know that you will wish to discard the formatting of the text you are pasting, you can use the special **Edit ► Paste and Match Style** command to do so. This will treat the text as though it were plain-text, and as such it will take on all of the characteristics of the text around the cursor position where it is pasted. Since the text is treated as plain-text, this means you will lose any inline formatting and function, such as hyperlinks and italics. If you intend to keep this level of formatting, deferring the problem to later by using the document conversion feature will be a better choice.

15.4.6 Preserve Formatting

If you are planning on letting Scrivener's compiler do all or most of the final formatting for you, it can often be useful to preserve ranges of text from the formatting engine. A common example of this would be the use of block-quoting as a left indent. Technical writers might find it useful to set apart blocks of text as "code", using a special font and layout. In the past, you had to break out these special sections into their own documents and preserve the formatting of the entire document. Now you can define these ranges right in their original context.

To specify a range of text for preservation, select the text in the editor, and then invoke the **Format ► Formatting ► Preserve Formatting** menu command. This will draw a blue dashed box around the text, which can be worked around and within like any other type

of formatting range, such as italics. Also like other formatting tools, to toggle a preserved range off, simply select the entire range and use this same menu command.

Preserved formatting will also protect your text when using the **Documents ► Convert ► Formatting to Default Text Style** tool, which is useful for homogenising the look of your document according to your preferences. Any text contained within these blocks will retain their original formatting and not be touched by this tool.

MultiMarkdown users can use preserved formatting ranges to define code blocks. Any paragraphs located within a Preserve Formatting range will have a tab character added to the front of each line within it, informing the MMD engine to treat the text like a code block.

Authors working with HTML may use Preserve Formatting to preserve ranges of raw HTML code. If you wish to type in some HTML code by hand and have it passed straight through to the final result, you can use this tool in conjunction with the HTML Settings compile pane (section 24.16) setting, "Treat 'Preserve Formatting' blocks as raw HTML", enabled.

15.5 Inline Images

While images themselves cannot be placed into the Draft as binder items, you can insert an image into the text either by dragging an image file in from the Finder; dragging an image document in from the Binder, Outliner or Corkboard; or by selecting **Edit ► Insert ► Image From File...**. The image will be placed at the current cursor position. Images placed into documents in this fashion will create a *new copy* of that image, even if it was dragged in from the Binder. This becomes important when working with placeholder images. If you intend to later edit your images or replace them with updated copies from a graphic designer, you may wish to instead use Linked Inline Images (subsection 15.5.3).

To resize the inline image, double-click on it. This will bring up the image scaling panel. Use the controls to shrink the image or increase its size, and then click on **OK**. Unchecking "Lock aspect ratio" allows you to adjust the width and height independently and therefore distort the image. Clicking **Cancel** restores the image to its former size and returns you to the editor. Note that some images types (notably, vector PDF) cannot be resized; double-clicking on such images will do nothing.

It is important to note that resizing images in the editor is an output, rather than an aesthetic, decision. When an image is resized in the editor, it will be directed to use that size in the final output as well. If you find the graphics you are using are too large, and are getting in the way of writing, you may consider using placeholder thumbnails instead.

Using linked images is ideal for working this way, as you can swap out the full-size images on the disk prior to compiling.

In the case of print-ready images, this often won't be too much of a problem as the text editor will display a rough approximation of the image's final size, rather than its actual pixel size, when the DPI and physical measurements are set to an appropriate value for printing. For instance a 300 DPI image that is meant to be 2.5" wide on the printed page will appear approximately 2.5" wide on your monitor, and so it will not dominate the editor.

For this reason, the primary unit of measurement used is points, rather than pixels, though these will be displayed below for your reference, as well as the current DPI of the image. Adjusting the scale of an image does not impact its literal pixels, but rather the point value which will be used to determine how large it displays. When targetting digital devices, 72 DPI is a good rule of thumb, though some experimentation may be required.

When using MultiMarkdown, inline images (and linked images for that matter) will be converted to MMD syntax, and the images themselves will be gathered into the compile folder for you.

> **Viewing inline images full-size**: To view an image contained inside text on its own, you can drag the image from the text into the header view of one of the editors. You can then double-click on the image and zoom in or out on the image. Note that any changes made here will not impact the embedded image in the text.

If an image is resized, or has alignment or line-spacing attributes applied to it, these options will be used in the final compilation. This is true even if the compiler has been set up to ascribe new formatting rules to your manuscript. In short, images come with a sort of limited Preserve Formatting (subsection 15.4.6) block implicitly around them.

To save the image to the disk, right-click on the image and select **Save as Picture...**.

15.5.1 Working with Print-Ready Images

If you've been provided full size, print-ready graphics for publication, it is often best to not embed them directly into your document. Embedding images in the document is good for small placeholders, but since the images are saved into the RTF file itself, adding very large high resolution graphics to your files may slow down Scrivener's ability to load and save your file while you work on it. If you need to work with print-ready graphics, read the following section on linking your images to the document, or alternatively specifying them with tags, rather than embedding them directly.

Additionally, the text engine is optimised to display graphics at their relative display size rather than their literal pixel size. An image that is set to 300 DPI that is four inches wide will display at a calculated size of 288 pixels, rather than the full 1,200 pixels, in the editor. This can impact the display size of the graphics in the output formats as well. Even though the graphics are not resampled, they will be displayed as they were in the editor, which may mean they will be displayed out of size with the intended output.

15.5.2 Embedding Inline PDFs

It is possible to drag PDF files into your editor to embed them. This feature is meant to allow the usage of graphics saved in the PDF format. It will not allow multi-page documents to be inserted into the final manuscript, and if the original PDF is pre-formatted for print, you will very likely need to crop the result down significantly so that it can fit within the page margins, using a PDF editor such as Preview, or Acrobat Pro.

When compiling to formats that do not support embedded PDFs (which is pretty much all of them), a raster graphic will be automatically generated for you and embedded as a PNG file, if the compile format allows embedded graphics. If it does not, such as with plain-text files, then as with standard graphic files, they will be omitted as per usual. When use **File ► Export ► Files...** and selecting RTF, this conversion will also take place. This will cause a loss of vector data, and so the processes uses a large PNG that is then shrunk down to scale to maximise quality. You can adjust the DPI use for this in the Import & Export preference tab (section B.10).

15.5.3 Linked Inline Images

Linked images are useful when the actual graphics in your project text are incomplete or placeholders for larger, production-ready graphics. If you are familiar with desktop publishing tools like Adobe InDesign, then the manner in which linked images work will be familiar to you. For those not aware, linked images are the placement of graphics in your text in such a way that the representation of the image in your editor is being generated by files *outside of the document*, as opposed to be saved as a part of the document.

To create a linked image, use the **Edit ► Insert ► Image Linked to File...** menu command. Linked images will look and act like embedded images in every way. You can by and large ignore the difference between them while working.

When compiling with linked images, the *current* version of those images on disk will be used to create the embedded copies used in the compiled version. Since nearly all of the compilation formats do not support active image linking, Scrivener must take the current version and embed it in the final copy. So if you are in a workflow that involves external help from designers, be sure to get your external images up to date before producing final compiled copies.

For increased performance, Scrivener uses a cached version of your image when you create a link to it while the project is open. Whenever you re-open the project, this thumbnail will be updated with the current image on the disk.

> **A Note on Links and Multiple Computers**: Those familiar with the use of linked resources will already know that using links means that the position of those external resources must remain stable, in order for the host program (Scrivener in this case) to maintain a connection with them. If the files are moved or renamed, the link breaks and you will need to manually update them. This means that in most cases, links will only work on one machine, since the full path name of a file tends to vary, given that it includes the name of the computer's hard drive. It is possible to carefully work around that, particularly with the use of external hard drives that are all mapped to the same drive assignment on the computer, or keeping all satellite computers identical in regards to user names and hard drive names.
>
> This problem becomes impossible to work around when working cross-platform, as the path addressing scheme used to name a file differs between Mac and Windows computers. This does not mean that using linked images prohibits cross-platform usage (or vice versa), but what it does mean is that only one machine can be your "home" computer in terms of final document assembly. A project can lose track of the linked images while you are working on other machines, but since these links are stored as text paths to the files, when you return to the home computer, they will all link up again.

15.5.4 Image Placeholder Tags

It can at times be desirable to not include the graphic in the source text at all, but rather to specify its name or file path, using text, similar to how one would link to an image in a web page or forum post. Textual links images afford you additional powers over typical images displayed in the editor. Since they are text based, they can be hinged upon compile settings, and thus you could images into one document format but not another if you pleased. You can also specify the dimensions of an image precisely, using text, which is important for getting them right the first time, if you do not intend to polish off the compiled document after compiling. Finally, image tags mean you can bring images to areas of the document that otherwise wouldn't support them—on account of their being defined by plain-text fields—such as custom separators, title prefix and suffix fields.

The basic syntax for the image tag is:

```
<$img:IMAGE_IDENTIFICATION;w=WIDTH;h=HEIGHT>
```

Everything after the image identification is optional.

Image Identification

The image must be specified either by its imported name, as displayed in the binder, or by a valid UNIX style path to the image's location on your computer. Path names can be relative to the location of the Scrivener project, so if the graphics and the project are stored in the same folder, you would only need to declare the path as 'image_name.png' for it to work. You can also use the UNIX shortcut for your home folder, the tilde. Full absolute paths are acceptable as well. Here are some examples

```
<$img:~/Dropbox/MyBookFolder/scene_break.png>
```

You might wish to use something like that, if you want to put a graphical break in between scenes, as a custom separator (subsection B.2.4). This also demonstrates a safe way to link to an image that will work from multiple computers all using this Dropbox account. They will also see a valid image at this path since the details of the user folder name are obscured by the tilde.

This references a graphic that has been stored in a sub-folder called Resources, which is a folder alongside the Scrivener project:

```
<$img:Resources/python_chart.jpg>
```

Here is an absolute link to an external hard drive called "Research", where a PDF in the sub-folder "MyThesis" in included. As with all embedded PDFs, you should ensure it is only one page in length and smaller than the text print block (sans margins):

```
<$img:/Volumes/Research/MyThesis/some_data.pdf>
```

When using graphics that you have imported into the project binder, you can refer to them by their binder title (you needn't know the extension or type of graphic it is). This will be used in precedence over any path names, and the first matching title name (counting from the top of the project) will be preferred over any other images likewise named in the project. A simple example, specifying an image that is called "Python Growth Rates in Everglades":

```
<$img:Python Growth Rates in Everglades>
```

If for some reason you have two graphics named "Python Growth Rates in Everglades", the image that is "highest" on the binder list will be selected. It starts at the top of the binder and works down until it finds a match. Likewise, if it finds any match at all it will use it, instead of looking outside of the Scrivener project for a graphic. So

if the binder name included the file extension, it would still use the imported version instead of a copy with an identical name located in the Scrivener project's location (making it a relative link).

Specifying Image Width and Height

The width and height of the image is an optional value, declared in pixels. If you have already sized the graphics correctly in accordance with your printer's specifications and output quality, then you will not need (or indeed want) to change the image size using Scrivener. However if you are not concerned with pre-sizing the images and just want to let Scrivener handle them for you, you can specify the width and height of the image in the placeholder tag, to instruct the compiler to resize it as necessary. It is valid to specify only the height or the width alone. This will cause the compiler to shrink or expand the image to match the specification you've provided, automatically resizing the opposing measurement, keeping the image's original shape, or aspect ratio, intact. If you declare both the width and height yourself, it is possible to squish or squash the image however you please.

```
<$img:Portrait of Dickens.jpg;w=650>
```

This will resize the portrait so that it is 650 pixels wide, and it will adjust the height of the image so that it remains proportional with the original in terms of aspect ratio; no squished faces here!

Tips for Usage

Here are some ideas for how this feature can be used, as well as some handy tips for common uses.

- *Common Width and Height*: you will need to know how many dots per inch your printer works with. It is not possible to know how many pixels you should use to make a picture full page height or width, until you know how many pixels it will be using to render an inch of image. You may need to contact your publisher to determine what the print resolution for the graphic plates will be. Once you know this number, you can multiply it by the width or height of the "text block", that being the area within the margins.

  A simple example would be US Letter with 1" margins, printed at 300 DPI. The width of the page is 8.5", subtracting two inches for the margins on each side, making the text block 6.5". That means $300 \cdot 6.5 = 1950$: 1,950 pixels wide.

- *Using Replacements to Change Images*: Replacements (section 24.18) in the compiler will be evaluated prior to any images being included. That means you could use

the feature to scan for the placeholder tag and remove it from the document, or perhaps change the path of the image to something else, which could be useful if you have two different groups of images, one optimised for black & white printing, and the other for an online full colour PDF or lower resolution e-book.

- *Searching for images*: since the image is defined as text, that means you can search for it by name, or even search for all "<$img..." tags in the draft to get a list of documents with figures, which could be saved as a dynamic collection (subsection 8.4.4).

- *Using images in the title prefix or suffix*: when compiling, it is possible to use an image in the prefix or suffix title area of the Formatting compile option pane (section 24.11.2), which means you can now do such things as adorn chapter titles with lines or other symbols.

- *Using images for section separators*: in a similar vein, you might wish to use a graphic to separate individual sections, rather than a string of text or blank space. It is possible to use image placeholders in the Separators compile option pane (section 24.9)

- *Specifying an image when two or more share the same name*: in cases where you have multiple images with the same name in the Binder, if you need to ensure that a certain one of these (rather than the first one in the list from top-down) is used, select the image placeholder and create a Scrivener Link (section 9.5) pointing to that image.

15.6 Using Snapshots

In essence, the theory behind a Snapshot is very simple. You can think of it as a way of manually saving your file for future reference. In a program like Scrivener which automatically saves your work as you work, you lose the benefit of being able to work "off the disk" for a while, and choosing at defined moments in time to either save what you've done, or discard it and return to what you have on the disk at the last save point. Of course, on the other hand, if your computer crashes (and as much as we try to avoid it, sometimes Scrivener crashes too) with an auto-saving program, you might only lose a few minutes of work, or even nothing at all, rather than everything you've done since the last save.

Snapshots help you bridge the gap between these two ways of working. It is like saving your document as you go, the only difference is, you get a *record* of each save point. That means you can go back and examine the evolution of the file and pick out just the right spot to restore from. In the course of editing, you might change your mind about something you deleted more than one save iteration ago. With a traditional save as you go

model, you'd be out of luck. With Scrivener, you can just go back through your Snapshot history, find the point in time where the deleted passage still existed, and copy and paste it back to the present—or even just roll back the whole document to that spot.

> **What About Milestones**: In addition to this usage, some people like to use Snapshots to set "milestones" as well as just capture periodic changes. Scrivener accommodates both styles of working, as you can take "Untitled" Snapshots with a single keystroke. You can also name your snapshots either when you take them, or later on. Named snapshots are easy to pick out of the crowd of Untitled ones, and it also makes it easier to clean up old routine Untitled ones once they've outlived their usefulness.

Theory aside, snapshots provide a way of backing up and saving different versions of individual documents. Taking a "snapshot" of a document takes a copy of the text of that document as it exists at that exact moment in time and stores it so that you can return to it, or restore it, at a later date. With routine usage, you need never need worry about making major edits to a document, because you can take a snapshot of it before you begin editing and then restore the older version later if you change your mind about the edits you have made. Snapshots only affect the *text* of documents (not notes, synopses or meta-data), and are therefore only available for documents that contain text (files and folders).

You can tell if a text document has any associated snapshots by its icon. Documents that have had snapshots taken of them have the top-right of the paper icon folded down in a "dog-ear". You can also tell by looking at the top of the Inspector pane if it is open. When there is an asterisk in the corner of the Snapshot button then it has Snapshots.

There are two primary ways to manage Snapshots: using the **Documents ► Snapshots ►** sub-menu, and the Inspector pane (section 19.8), the latter of which will be covered completely in the referenced section. You can also conveniently use the **⌘–5** keyboard shortcut while editing a document to take a quick snapshot in the background.

15.6.1 Creating Snapshots

While working on a document, there are three easy ways to snapshot it for future reference.

The key thing to understand about Snapshots is that they provide a way to set save points for *individual* items in the binder. They are not a tool for providing an overall snapshot of the entire project, structurally speaking, and are probably not the best tool for taking a quick snapshot of your entire draft. In most cases, large scale backups like this would be best created using the **File ► Back Up ► Back Up To...** menu item, or by duplicating large portions of the Binder and stashing them elsewhere for future reference.

That aside, it is simple to quickly snapshot multiple documents.

1. Select the documents you wish to snapshot in the binder sidebar, corkboard, or outliner

2. Press ⌘–5, or **Documents ▸ Snapshots ▸ Take Snapshots of Selected Documents** to create a snapshot into each item's individual snapshot list.

   Snapshots can later be accessed in the Inspector pane. You can jump straight to them with ^⌥⌘–M, or **Documents ▸ Snapshots ▸ Show Snapshots**.

3. The alternate way of taking a snapshot is to supply them with a title as they are created, use **Documents ▸ Snapshots ▸ Take Snapshot with Title**, or ⇧⌘–5.

   This can be done for single or multiple selections as well. In the latter case, the title will be added to each document in the selection. Snapshots can also be titled at a later time using the Inspector.

> **Extensive Annotation**: A trick for more thoroughly commenting on the purpose or state of a Snapshot can be to combine it with the use of inline annotations. Consider writing a short synopsis of what the snapshot was taken for at the top of the document before snapshotting, and then removing it afterward. Now that statement will always be clearly presented to you at the top of the Snapshot text preview.

The third method for creating snapshots is from the Snapshot pane itself in the Inspector. Click the **+** button in the header area of this pane to create a new snapshot.

15.6.2 Managing and Comparing Snapshots

Existing snapshots are all managed in the inspector pane. To jump to snapshots, click the button with the camera in the Inspector footer bar or use the **Documents ▸ Snapshots ▸ Show Snapshots** menu item (^⌥⌘–M). In the snapshots pane, you can create new snapshots from the current text with the **+** button, remove old snapshots by selecting them in the snapshot list and clicking the **-** button, or roll back to a prior version of your text using the **Roll Back** button.

More details on the functions available in the inspector can be found in Snapshots Pane (section 19.8).

15.6.3 Comparing Changes

The comparison feature lets you analyse the differences between versions of your text. There are two ways of using this:

1. Compare with current: select the snapshot you wish to compare against the current text, and then click the **Compare** button. The changes will be highlighted in the text preview area.

2. Compare two snapshots: select two snapshots in the table and then click the **Compare** button. The differences between those two versions will be highlighted.

While in comparison mode, the text preview area will switch to plain-text so that only actual changes to the content can be analysed. To return to standard viewing mode, click the **Original** button.

Comparisons can be reviewed in the standard editor, and are thus useful in conjunction with a split view, as you can review the edits alongside the current version of the document. To view a snapshot in an editor split, drag the snapshot from the table in the Inspector to the header bar of the split you wish to load it in. If you hold down the Option key while doing this, comparison mode will be engaged for the split.

15.6.4 Viewing Snapshots in the Editor

It is possible to load snapshots as read-only text into the main editor splits. To do this, simply drag the snapshot you wish to view into the header bar for the editor split you wish to load it in. You can also load the snapshot in comparison mode by holding down the Option key as you drag the snapshot into the header bar.

15.6.5 Automatically Created Snapshots

There are two ways in which snapshots will be optionally created for you:

- *Synchronisation*: the various tools for synchronisation in the **File ▶ Sync ▶** submenu are set by default to take snapshots of any documents that are changed (either on the server or in Scrivener). When this sort of action is taken, the snapshot will be titled appropriately with a descriptive parenthetical regarding why the snapshot was taken.

- *Automatic Snapshots on Manual Save*: not enabled by default, this "Take snapshot of changed text documents on manual save" option, in the General preference tab (subsection B.2.2) will create a snapshot of each binder item with text content that has been altered during the current session. This is only considered for changes made to the text content itself. Since snapshots do not save meta-data such as the synopsis or notes, changes made only to these text fields will not trigger an automatic snapshot. Snapshots taken for this reason will be marked as "Untitled (Save)".

15.7 Auto-Completion

There are three forms of auto-completion in Scrivener:

Character substitution Adjustments made to the characters as you type them in, such as setting quotes to typographer's "curly" or "smart" quotes, as they are alternatively called; or adjusting stand-alone i's at the start of a sentence to capital form.

Word and phrase completion As you type, you can have Scrivener suggest words for you either automatically, or with a shortcut (the default); custom phrases can be added to each project, such as proper nouns or frequently used scientific terms. Separate from the project phrase list, you can also auto-complete based on existing title names from the binder.

Scriptwriting abbreviations Common scriptwriting abbreviations such as "O.C.", "Ext." and so on, can be suggested as you type. These are generally defined in the script formatting definition (**Format ► Scriptwriting ► Script Settings**).

15.7.1 Character Substitutions

With character substitutions, you can set which characters will be replaced as you type. Scrivener uses a combination of built-in enhancements, plus OS X's (Snow Leopard and greater) built-in substitution engine. These can all be set in the Corrections preferences tab (section B.9). Each option is documented more fully, there.

These adjustments always happen automatically as you type, but some require you to terminate the word you are typing in, before they will take action.

15.7.2 Custom Auto-completion

Word and phrase completion can be used to suggest completions as you type, but more commonly you will use a keyboard shortcut to trigger the completion menu. ⌘–., ⌥**Esc**, and **Esc** (all by itself) work on most systems, but on some only one will work. In particular, ⌥**Esc** is frequently set to open Front Row on most Mac laptops.

Each project has its own custom auto-completion list, which adds words and phrases to the stock dictionary that is provided. There are two ways to add items to this list:

1. The menu item, **Project ► Auto-Complete List...** and use the **+** button to add new lines to this table.

2. Select the text word or phrase you want to add, in the editor, then right-click on it and select **Add Selection to Auto-Complete List**.

If you would like for Scrivener to suggest custom phrases to you as you type, set "Suggest completions as you type" in the Auto-Complete preference tab, and then disable "In script mode only"; making sure the third option is enabled, otherwise the entire language dictionary will be unlocked and most will find this to be too intrusive.

> **Copying completions between projects**: If you wish to transfer custom completions from one project to another, you can simply open both projects simultaneously, access their respective Auto-Complete lists, and drag and drop the terms from one panel to another.

15.7.3 Auto-Completion Scope

When using the scriptwriting feature of Scrivener, key words and phrases can be automatically added to the auto-complete list where appropriate. For example, if you type in the name of a character in the Character element, it will automatically be added to your project's auto-complete list.

By default, these special automatic entries will only be made in reference to the element they came from. A character name, to continue that example, would be defined in the auto-complete list as being applicable only to the Character element type. It will not appear as a suggestion when you are typing in the Scene Heading element. So by and large you will never have to bother with setting anything up here. As you write, Scrivener learns about your script and becomes better at making suggestions to you depending on where you are and what you are typing.

If you want to make changes to the scope of a completion, so that it is present in different or even all element types, this is possible in the **Project ► Auto-Complete List...** window. In the Scope column, you can change the element type or assign it to. There are two special settings available:

- *General*: The default when adding your own to the list by hand. Documents set to use scriptwriting will ignore entries of this scope (excepting text set to "General" or "General (Centered)").
- *All (Text & Scripts)*: Similar to General, only these entries will absolutely be available everywhere, even in scripting elements that maintain their own lists.

If a script element has been set to not use the project auto-complete list in its script settings, then it will ignore the entry, no matter what scope settings you use.

15.7.4 Binder Title Completion

If you have started typing in the name of a title that is found in your binder, you can request a list of completions from Scrivener by pressing a different shortcut, **^Esc**. A list

of all the titles that are still candidates for a match will be presented. As with the above, you can use the arrow keys to select and item and hit return to insert the full title.

This feature is particularly useful in conjunction with the "wiki style" Scrivener link option. Simply type in ((start typing in the title name, hit ^Esc, select a title, and then close with)). A Scrivener link will be automatically created for you.

15.7.5 Scriptwriting Auto-Completion

Scriptwriting mode, by default, engages a more aggressive auto-completion method, which scans as you type and looks for completions, matching the behaviour of most scriptwriting programs on the market. Most of the script formats that are shipped with Scrivener come stocked with many common phrases and abbreviations, which will be contextually tied to the element you are currently working in. For example, "MOMENTS LATER" will appear whilst typing in a Scene Heading element, but will not appear in a Dialogue element.

Most script formatting settings will also automatically add anything typed into certain key areas of some elements into the project's master auto-completion list. Character names, locations, and so on will be checked for and added, making rote data entry more efficient as you write.

Since auto-completion cannot reliably determine appropriate letter case, sometimes it may produce results that are undesirable, such as a mixed case location name in a slugline. To remedy this, when selecting the auto-completion you wish to use, hold down the Shift key. Any word or phrase selected with the Shift key held down will complete using uppercase letters.

15.8 Text Bookmarks

Text bookmarks are useful when you wish to quickly mark a spot in your text in a way that can be easily recalled later on, or for marking out sections of text in longer documents. There are two types of bookmarks:

1. *Bookmark Annotation*: This is your standard bookmark. Great for quickly marking a spot for later editing, quick navigation, or whatever you require.

2. *Bookmark Header Annotation*: Will create an offset label in the bookmark navigation menu, indenting regular bookmarks beneath it until another header style bookmark is encountered. Useful for documents with many bookmarks.

Bookmarks work by placing a special marking at the beginning of a paragraph or line, and there are two tools for doing this without disrupting the writing process. Use the

Edit ► Insert ► Bookmark Annotation menu command, or ⇧⌘–**B** to place the marker at the front of the current paragraph without moving your cursor.

Use **Edit ► Insert ► Bookmark Header Annotation** to insert a header style annotation in a similar fashion as a standard annotation.

> **Custom bookmark labels**: Bookmarks are simply an inline annotation starting with an asterisk and a space. The first few words of the line following will be used to label the bookmark in the navigation menu. If you wish to provide a custom label, add it to the annotation after the asterisk and the rest of the line will be ignored.
>
> The same holds true for header style bookmarks. These use two asterisks and then a space, so to give them a custom label, enter the label into the annotation after these asterisks.

In addition to formal bookmarks, any inline annotations that appear on a line by themselves will be added to this menu as well, regardless of whether or not they have an asterisk in front.

15.8.1 Navigating to Bookmarks

Once bookmarks have been created, they can be viewed in the header bar icon menu, under **Bookmarks**, or **View ► Text Bookmarks** menu. To navigate to a particular bookmark, select it from this menu.

When viewing several documents in a Scrivenings session, all of the bookmarks found within the individual documents will be used to populate this menu.

To remove a bookmark, simply delete the inline annotation on the respective line.

15.9 Editing Multiple Documents

The "Scrivenings" view mode (**View ► Scrivenings**) allows you to edit multiple text documents as though they were one long document. This way, you can write in small chunks and then combine them in any way you like to see how they work together. For instance, you might write lots of small scenes for a chapter, and then edit them all together in an Scrivenings session to see how the chapter works as a whole.

Scrivenings mode is only available when the entire selection (or contents of the selected container) are text or folders.

15.9.1 Viewing Multiple Texts as One Document

Scrivenings mode is, like corkboard and outliner view modes, another way to view multiple selections of documents. How you view more than one document changes, depending on your approach. Viewing more than one document can be as simple as clicking

on a folder. By default, this will view the items within that container as index cards on a corkboard. However using the Group Mode segmented control, you can easily switch your view over to Scrivenings, the left-most icon showing a stack of papers. Another way of switching is to use the View ▶ Scrivenings menu command, or pressing ⌘–1 on your keyboard. As with corkboard or outliner, you can also form scrivenings sessions by selecting multiple items in the binder, or clicking on the header bar of a collection or search result.

Unlike corkboard and outliner, scrivenings can be taken into composition mode. Simply press the Compose button in the toolbar after setting up the session. Read more about composition mode (chapter 17) for general information on this feature.

You can tell when you are in Scrivenings mode by the horizontal divider lines that will be drawn between each section represented in the session. Meanwhile, the header bar will state the name of the currently viewed container followed by "Composite" in parenthesis—or "Multiple Selection" followed by the title of the document you are currently editing *within* the session. As you switch from document to document, the inspector will follow along so you can see which document you are in by glancing at the index card in the inspector.

Scrivenings sessions are one of the few areas in Scrivener that consider your text from a "flat" perspective. There is no indication of depth while editing documents. A prologue at the very top will be printed right along with items which are buried in sub-folders. This is intentional, so you can concentrate solely on the same flow of text that your readers will use.

The footer bar will switch to an alternate mode which disables the per-document goal meter (subsection 14.2.3) as it no longer has any meaning in a multiple document setting. Additionally, the word and character counter in the footer will count the entire session, making Scrivenings a great way to quickly view the word count of a particular section of the Binder.

If you would like for titles to be displayed more prominently, you can enable them by switching them on in Format ▶ Options ▶ Show Titles in Scrivenings. The appearance of these titles will be dictated in Formatting preferences tab (section B.8). These titles can be edited in the same fashion as editing any other piece of text in the editor (though formatting will not be allowed).

In that same section you can opt to use a slimmer divider which uses no vertical space to draw itself. This will be of most use to those who must work with accurate page counts in Page View, or simply as an aesthetic preference.

15.9.2 Editing with Scrivenings

For the most part, editing in Scrivenings mode is just like editing single documents one at a time, thus most of the tools that you have learned so far will be available to you. You can select all of the text and format it, copy and paste it to another application, create snapshots of individual portions of the session, and so on.

Due to the manner in which multiple real files are being combined as though they were one "file", it is not possible to select across a file boundary with the keyboard, or to delete text from across multiple files at once.

These limitations are unavoidable. If you were to delete a range of text that spanned two or more documents, Scrivener would have to guess where one document now ended and the other now began—and if the whole of the text of one document had been deleted in the process, Scrivener would have to decide whether to delete that document, too. These are all decisions that are better left to you.

Since Scrivenings is a full view mode, it affords a great degree of latitude in how integrated it is with the outlining process, especially with a locked editor (⌥⌘–L).

- While editing, new documents can be created with ⌘–N, inserting them below the current document that is being edited; likewise with splitting documents.
- With titles turned on, it is possible to create new items without ever switching back to the Binder, making this combination of actions one of the most seamless ways to flesh out your outline while you write.
- Moving documents around in the Binder will automatically update the order in which they appear in the Scrivenings session.
- Deleted documents will be removed from the session.
- Merging documents from the Binder will make this change in the session

15.9.3 Useful Tips

Navigating with the sidebar navigation buttons If you click one of the navigation buttons (subsection 14.1.4) in the header bar, and that navigation action will result in movement within the boundaries of the active session, you will be navigated from section to section without disturbing the session. If you leave the boundaries of the session, or if the original selection was the result of a non-linear multiple selection, it will be dismissed.

Using the Go To menu to navigate While the active split is in Scrivenings mode, the header bar's item icon menu will display only the contents of the current Scriven-

ings session. Selecting items from that list will jump you straight to that document like a small table of contents.

Selecting only the document you are editing Use ⌥⌘–A to select the entire text area for *only* the document you are actively editing within the session. Standard ⌘–A will always select the text of the entire session. Selecting by section is also a useful way to get a quick word count of the active document in the footer bar.

Jumping to the end or beginning of a section In conjunction with the above tip, a neat trick for jumping to the very beginning or end of the current section is to select the section with ⌥⌘–A, and then use the LeftArrow to jump to the beginning, or the RightArrow to jump to the end, as this is standard behaviour for any selection.

Isolating the current document If, in the course of editing a scrivenings session, you find you wish to isolate the current document so you can focus on it exclusively, you can simply tap ⌘–4, or use View ▸ Go To ▸ Editor Selection. To return, you can use View ▸ Go To ▸ Enclosing Group, or use ^⌘–R.

Lock in Place navigation If you are in an Edit Scrivenings session and lock the editor containing the Scrivenings view in place (using "Lock in Place" from the View ▸ Editor ▸ Lock In Place or header view icon menu), clicking on documents in the sidebar that are in the Scrivenings session will move the insertion point to the beginning of that document in the editor (just like Go To).

Meta-data grouping Some forms of meta-data will group themselves when using scrivenings. footnotes and comments (section 18.2) will be displayed all together in the inspector in a single large stack, allowing you to quickly see if any of the constituent articles within the session have notes. Text Bookmarks (section 15.8) will also be grouped together in their access menu.

Vertically accurate scrivenings divider If for reasons of taste, or formatting necessity, you require a zero-height scrivenings divider, you can toggle on a slim mode in the Formatting preference tab (subsection B.8.3). This mode will draw small "crop marks" in the margins instead of using a divider across the middle. If vertical accuracy is a necessity, you will also want to avoid usage of titles in the scrivenings session. When in slim mode, the height of the page should be preserved for purposes of rough preview with the Page View editing mode.

Viewing only items which can be compiled By filtering items in views (section 9.9), you can create a Scrivenings session that includes only items which can be compiled with a single click. This can be exceptionally useful if you tend to write notes into

your drafting area, and set those notes to not be included in the compile. To use this feature, Option-click on the container (including a Collection or Hoist header bar) you wish to filter, or with a selection of items, use the **Documents ► Open ► with Compilable Subdocuments** menu command.

Chapter 16

Page View

By selecting **View ▸ Page View ▸ Show Page View** while editing a text document, you can transform the visual presentation of the editor to using a virtual page, which can optionally show two opposing pages at once, with **View ▸ Page View ▸ Two Pages Across**. Since Scrivener does not keep track of actual post-print layout pages, this should not be relied upon as a full-spread layout preview, as the even/odd arrangement you see in the editor might very well end up being swapped in the final product.

In fact, for most uses, Page View is for simulating the look and feel of writing on real pages, and is thus an aesthetic preference, not a print preview tool. In some cases, especially where the compiled product will look identical to the formatting you see in the editor, it can be used as a fairly accurate gauge of writing progress in terms of literal pages. Read on for tips on the best ways to set up this feature for this style of working.

Editing in Page View is otherwise identical to use the standard drafting style editor in all ways, and will work with Scrivenings mode, too. The two optional editor features, Typewriter Scrolling (subsection 15.3.4) and line numbering, will be unavailable while using this editor mode.

16.1 Page View Dimensions

Page View will place your text in a to-scale representation of your project's print and margin settings, found in the **File ▸ Page Setup...** dialogue. These settings will be used regardless of your compile settings, giving your the option to use a different paper shape for the screen than you would use for print.

16.2 Tips for Accuracy

When used in conjunction with export fonts, formatting and accurate page dimensions, the resulting page estimate (which will be calculated in the footer bar statistics area) can be quite close to the actual end product, and thus will be of considerable use to anyone who requires pages as a metric, such as scriptwriters. Do note however that page numbering will always be relative to the section of text you are viewing. It is not intended to be a method of finding "page 83" from a stack of printed out papers by your desk. It would

be computationally prohibitive to provide this information in real-time, based on the fact that Scrivener is fundamentally not a "What You See is What You Get" editor, like a word processor.

To increase page count accuracy in a scrivenings session, you should use the "Separate with single line breaks" option in the Formatting preferences tab (subsection B.8.3). This alternate method for showing the boundaries of documents in scrivenings mode uses no height, and so will not vertically distort the size of the session. The use of the titling feature in scrivenings should also be avoided as they will add extra height that likely will not be in the final composite. This method cannot be used with any content that requires a lot of post-compile layout, such as footnotes, as footnotes will be added to the total height of the page and change the flow from what you see in the editor.

Finally, you should ensure that your **File ► Page Setup...** settings match your compile settings, so that what you see in the editor matches the shape of the document that will eventually be compiled.

Chapter 17

Composition Mode

Composition mode allows you to edit your text in a distraction-free environment. Made popular by a plethora of "zen-ware" applications which promise a stripped down and often plain-text writing view, Scrivener's approach is decidedly more expressive—though you can keep things dirt simple if that is what you are looking for. You can view any text document (or Scrivenings session) in Composition mode, with a virtual page treatment on a background (by default, black). Meanwhile you will have full access to your document's meta-data in a floating low-impact Inspector panel, as well as the ability to view QuickReference panels for purposes or research availability and other referencing. Ultimately, you can compose and refer to your notes without the noise of the rest of your desktop or even Scrivener itself. Consequently, you'll find that Scrivener "disables itself" to a degree, while in Composition mode. Menu commands that would otherwise be available are turned off—in most cases because these are commands which would have no meaning at all, such as commands for merging two selected items in the binder, or enabling corkboard view.

To determine what will be loaded into Composition, Scrivener uses the active selection. This selection can be made either in the sidebar, or in an outliner or corkboard view. This works by implication as well: when no cards are selected in a corkboard, technically the *container* is selected, as can be witnessed in the inspector, and so it is the container's text itself that will be loaded in Composition, which might often be blank.

There are two exceptions to this rule:

1. *View as Scrivenings*: If you are viewing a group of documents in Scrivenings mode, then Composition mode will use that same mode. This is the easiest way to load an entire container's text into Composition mode, as it requires no further selection.

2. *Pre-load documents into history*: If multiple items are selected and Scrivenings mode is not enabled, then each item will be loaded into the Composition mode's history queue. Working in this fashion, you can use ⌘-(and ⌘-) to flip between the documents without returning to the main project window.

Launch Composition mode either by hitting ⌥⌘–F, or select View ► Enter Composition Mode.

When you first enter Composition mode, you will see your document in the middle of the screen set to a warm-grey paper colour, and masking the rest of the screen, a black background. Nearly every aspect of this experience can be changed, some right within Composition mode. You will also briefly see the Control Strip (detailed below). To retrieve it after it has disappeared, slide the mouse pointer down to the bottom of the screen and let it sit for a moment. So long as the mouse remains within the control strip, it will stay visible. Once moved back up, the control strip will disappear out of view.

The menu bar can be accessed by similarly sliding the mouse pointer to the *top* of the screen. After a momentary pause, the menu will appear. This is a useful way to access advanced formatting tools, and even navigate around in your project using the View ► Go To ► sub-menu without leaving composition mode.

17.1 Using Background Image and Textures

You can replace the default solid background colour with an image of your choosing, by using the View ► Composition Backdrop ► sub-menu. The image will be scaled to fit your screen using the maximum height or width, whichever is better. This can mean that if the image does not match your screen's aspect ratio, the background colour may still be visible along the sides or top and bottom.

Textures can be assigned to the background area on either side of the "paper" or the paper itself behind the text. Textures are graphic files you select which will be tiled to fill the entire visible area. Textures can be configured in the Appearance preference tab (section B.3), as an additional option in the customisable colours section.

17.2 The Control Strip

The Composition control strip contains various controls for manipulating the interface elements available to Composition mode. All of these settings operate on a project-by-project basis, as opposed to the global scope of the application preferences.

It is possible to leave the control strip open perpetually, by leaving the mouse at the bottom of the screen. If you wish to keep track of your word count, or are working in script mode, this can be a handy way to keep tabs on your status. If you do so, you may wish to set the paper height (detailed below), so that the bottom of the page does not fall below the controls.

In order from left to right:

Text scale This operates in an identical fashion to the text scale zoom in the standard editor. The only difference is that Composition mode stores its own scale setting, independent of the editor splits.

Paper Position Select where the "paper" (the column of text) should be anchored on the screen: on the left, in the centre or on the right. You may want it in a non-central position, for instance, if you have a lot of notes to which to refer and you want the inspector open (and wide) next to the text.

Paper Width Use this slider to change the width of the text. You can set it so that the text takes up the whole width of the screen or appears as a column.

Paper Height To access this slider, hold down the Option key. It will replace the Paper Width slider, from above. This sets how tall the paper column should be, giving you a "letterbox presentation" at its more extreme settings.

Keywords Brings up the standard Project Keywords (section 10.3) which you can use in conjunction with the inspector (below) to assign keywords to the current document.

Inspector Brings up the floating Inspector panel (see below).

Go To Operates as a handy navigation tool, and functions identically to the main View ▶ Go To ▶ menu. Using this, you can switch between documents without returning to the main project window. You may also use the history shortcut keys, ⌘-(and ⌘-) to navigate to and front documents you've visited while in composition mode. Composition history is separate from the main editor history—it will not add to it and when you first start composition mode, history will be empty.

Dynamic Controls & Statistics The middle portion of the control strip changes depending upon your editing mode. In standard editing mode, this will display the word and character count for the document. As with the standard editor, this will also display the counts for any selected text, using a blue label.

In scripting mode, this portion of the strip will display element hints as well as the element selection menu (which you can always call up with ⇧⌘-Y, even if the strip is not visible).

Background Fade Fades the background (everything outside of the text paper column) in and out. Useful if you want to refer to material in other applications (or the main project window) in the background whilst remaining in composition mode.

Paper Fade When a backdrop image is in use, the "Background Fade" slider will be replaced with this one, which operates in an identical fashion, only it lets you blend the paper colour with the backdrop, or reduce its visibility entirely.

When both a background texture or backdrop and a paper texture are in use, this slider will be entirely disabled.

Exit Composition Mode The button at the far right will exit Composition mode. You may also press the **Esc** key, or **⌥⌘–F** to return to the main project window.

17.3 Floating Inspector HUD

The composition mode version of the inspector gives you near full access to the inspector data which will float over the editor so it doesn't get hidden behind the background. Use the top-drop down menu to access the different data panes. Label and Status will always be visible at the bottom of this window. For a full discussion on these views and what they represent, read Meta-Data Types (section 10.1), and Inspector (chapter 19).

The top selection drop-down will provide you with the following options:

- Synopsis
- Picture: If a photograph has been attached to the index card, you can view it with this option.
- Document Notes
- Project Notes: If you have created additional document notepads, they will all be listed here.
- Keywords
- References
- Custom Meta-Data
- Comments & Footnotes

Each of the individual panes will be provided with a full complement of editing controls, so you can add and remove items from these lists, just as you would with the standard inspector.

17.4 Customising Appearance

Appearance settings can be set in the Compose preference tab (section B.5). Refer to the documentation provided for this pane for greater detail on available options.

17.5 Composition Mode with Multiple Displays

When using two monitors, you can configure Scrivener to use the secondary display as the composition window instead of the primary display. This can be adjusted in preferences.

Also, if you don't mind taking a performance hit, you can choose to leave the main Scrivener interface open on the primary display. When doing so, Option-clicking on text or folder items in the Binder will load these items into your live composition session. As with the standard editor, using the Command and Shift keys will select multiple documents and form a Scrivenings session. To utilise this in the active composition window, just add the Option key to these modifier keys.

Chapter 18

Annotations and Footnotes

As a tool designed for the production of texts, there are many methods for facilitating not only the writing process, but the editing process, and the "public notation" process which is typically expressed via some form of page or section footnote, or endnote at the conclusion of the work. This chapter will cover all of the various methods that Scrivener provides. We will cover:

- Annotation and Commentary: the ability to place production notes within the document, making it easy to communicate changes or needs to yourself, collaborators, your editor, or to receive the communications from others. Scrivener supports the RTF standards for commentary, and will convert any word processor applied notes to your preferred notation style, which can be configured in the Import & Export preference tab (section B.10).

- Footnotes and/or Endnotes: being a drafting tool, Scrivener makes no attempt to typeset and number either of these as you write, however it is easy to insert notes precisely where you need them, and when you compile they will presented according to your exacting demands. As with comments, RTF standards are supported and incoming notes will be converted to your preferred notation style. Scrivener supports two note streams, allowing you to output both footnotes and endnotes, if required. In the case where specific formatting is required, it may be advantageous to use a citation manager that can scan an RTF and produce a bibliography for you (section 21.6).

- Text colours and Highlighters: to aid in increasing the visibility of passages, you are provided with an extensive complement of tools for highlighting and marking your text in colour.

- Revision markings: the automatic application of text colour, as you edit and write, can be assigned individual colours for up to five revisions, making it easy to make sure your colleagues (or yourself) know what you have changed and vice versa.

Marking your text is one thing, finding those markings amongst a 100,000 word draft is another. The robust Find by Format and Text Function (subsection 21.1.6) makes it easy to quickly step through your draft, point by point, isolating and addressing issues or reviewing changes that you've made.

Annotations and footnotes (this document will refer to all reader notes as footnotes, even if their intended result will be endnotes) come in two different flavours. Inline notation, which are added directly to your text, and linked notation, which are attached as a special form of "link" to the base text of your book, with the content of the note stored off to the side in the inspector, much as in many word processors. Which of these to use will remain up to you and your preferences in most cases. You might even find that a mixture of methods will suit you, using inline notation for short comments to yourself about prose, and linked notes for other types of comments. There are advantages and disadvantages to each method:

- Inline notes are always visible in your text; there is no way to diminish their prominence. So for some forms of notation, this can be an advantage in that you cannot defer or easily ignore them. This also makes it easier to see your notes and your book at the same reading speed—there is no need to look off to the side to get a feel for the "meta" book.

- Linked notes do not disturb the flow of text, no matter how large they may be. This means even the lengthiest of notes can be placed into your text without having to scan from word to word in order to read the underlying book text.

- Inline notes, being within the text itself, do not require any additional interface to use and never require the mouse to read. They thus work well with a slim workflow, or in situations where screen space is at a premium.

- Linked notes can act like bookmarks. Clicking on them in the inspector will whisk you right to the spot in the text where they are anchored.

- Inline notes can be placed anywhere you like (especially annotations), even in between paragraphs or at the very beginning or end of a section, whereas linked notes require something to "anchor to". This makes them more useful when jotting down notes in sections before you've even started to write.

- Linked notes can be easily viewed together in a collected interface no matter how far they are spread apart in the document. This becomes especially advantageous in a Scrivenings session. A note on page 10 has the same prominence as a note on page 1.

- It follows then that inline notes are only visible in the current contextual surroundings. Notes that are pages away, and do not presently concern you in your writing and editing, are hidden by the same virtue that hides these irrelevant texts from you: the sheer bulk of your words.

- Inline notes exist *around* your text and do not depend upon it to exist. This means you can edit the text that the note refers to freely, without worrying about losing your comments. This is in opposition to linked notes, which are anchored to the text—if the text is removed or entirely altered, the note will be deleted. By corollary, inline note quantities can exceed the textual capacity of the base text. For most authors this will not become an issue, but in some fields, such as qualitative data analysis, where the amount of annotation can exceed the text, running out of suitable anchor text could become a problem with linked notes. Inline notes, being unanchored, have no limits.

These are just a few examples, and hopefully that gives you an idea of the individual unique merits in these formats. In some special cases, using all four methods together will afford you with additional compile powers, such as handling multiple note streams, or using annotations as "functional" text and linked comments strictly as notes. The next few sections will address each of these types in more depth.

18.1 Inline Notation

Scrivener sports a unique method for taking notes and indicating footnotes, by placing them directly into the main text stream itself. It mimics the natural annotation process that one applies when working with paper, where notes are scribbled below the problem lines themselves, into margins, or anywhere else they will fit in. It means your notes remain prominent and easy to find so long as they remain issues worthy of attention. The same treatment can be given to footnotes as well. Keeping the actual text of the footnote directly in-line with the text it augments can increase productivity and help keep your meta-book relevant and cohesive. Further advantages are in entry speed. Since notation is done directly in the text while you are typing or editing, it is easy to flow from "editor" to "writer" without disrupting the process with floating windows, margin bubbles, or the many other mechanisms that applications use to attach notes.

Despite the advantages, there are still cases and tastes which would prefer to keep notation out of the main text area. If you have interest in commenting on your work or adding footnotes to it without adding text to the main editing area, please skip down to Linked Notation (section 18.2).

> **What They Are**: Inline notation is fundamentally a form of formatting, just like any other formatting spans in your document, such as bold or italic. Yes, Scrivener has tools for working with notations in unique ways, but they will respond to many of the actions that are designed for working around formatting, such as Paste and Match Style. When cutting and pasting text into and out of notations, try to think of them as analogous to bold text. To paste some external text into a note, you'll need to use Paste and Match Style for the text to acquire the notation formatting. The same is true going in other direction. Additionally, ranges can be toggled on and off for a section of text, just as you could with an italic range.

How you interface with inline annotations and footnotes is identical at the writing phase. Since inline notation works just like ordinary formatting, creating a note or footnote from existing text is as easy as selecting the range of text you wish to adjust and clicking on the appropriate menu command or toolbar icon. To create an annotation, you can use **Format ▶ Inline Annotation** or press ⇧⌘–**A** on your keyboard. As with other formatting tools, this works in a toggle fashion. If you select some text that is already set to be an annotation and use this command, it will be returned to regular text. For footnotes, everything is the same; just use **Format ▶ Inline Footnote** or ⇧⌘–**F**.

To create a new footnote or annotation from text that does not yet exist, you can use these same commands to start typing in the desired notation mode wherever your cursor is placed in the editor.

Moving notes around is a simple matter of cutting and pasting the text itself. Since notations are formatting, they will move along with the text no matter where you take them, unless you use Paste and Match Style.

18.1.1 Inline Annotations

Annotated text will appear red with a "bubble" around it, by default. You can change the colour by placing the cursor anywhere within it and using the colours palette, which is toggled via **Format ▶ Font ▶ Show Colors**, or ⇧⌘–**C**. On a per project basis, Scrivener will remember the last colour you chose and use it for the next new note you create (thus you can effectively change the application default to another colour by creating your own starter template). To split an annotation into two pieces, select the portion you wish to split off and assign it a different colour.

To ensure your text exports as expected, you must check that the text surrounding your annotation would make sense were the annotation not present, including how you set out your whitespace (spaces, newlines etc). In other words, if you place an annotation between two sentences, make sure that there is only one space (or two if you use two spaces between sentences) separating the sentences outside of the annotation. On

the other hand, when you choose to compile annotations into the document inline, all spacing within the annotation will be preserved; the end result will look like it does in the main editor.

Since you can choose to omit annotations from the text when you export or print it, they are very useful for jotting down notes and ideas inside the main text that are not really part of the main text, but which would be better placed next to the section of text to which they refer, than in the Notes pane. When you export to RTF format, annotations can also be converted to RTF comments that appear in marginal boxes in compatible word processors, rather than omitted, as well as a few other display options for other word processors or formats. For details on what options are available for omission or conversion during compile, see Footnotes/Comments (section 24.21).

While it is possible to use annotations to create footnotes, this compile-time conversion has fewer options available to it than standard footnotes. For fully featured footnotes, continue to the next section.

> **Important note**: Although you can create tables and bulleted lists inside annotations and footnotes in Scrivener, they are not supported inside annotations and footnotes in the RTF, RTFD and Word exporter. This means that if you create tables or bulleted lists inside annotations or footnotes in Scrivener, when you export to RTF you will find that the tables and lists are stripped out. This is because certain limitations in the OS X system can cause crashes or strange results during the RTF conversion process in situations where tables and bulleted lists clash with the boundaries of annotations and footnotes. If you require such elements, it is thus recommended that you format them at a later stage of the draft in a dedicated word processor.

If you would like to select the entirety of an inline note, simply right-click anywhere within it and choose “Select Annotation” or “Select Footnote” as the case may be. This command is also available from the Edit menu, and will select the note that the cursor is currently within.

18.1.2 Inline Footnotes

The creation of inline footnotes is quite similar to annotations, and really in most ways, footnotes act identically to annotations. The main difference lies in the fact that annotations are meant to be omitted upon export or print-outs (though do not have to be) whilst footnotes become “real” footnotes (or endnotes) upon export, and thus have more options available to them for styling footnotes, particularly with the RTF format option. Visually in the editor, the difference is that footnotes are not coloured but are surrounded with a black line and have a grey background.

> **Formatting Footnotes**: Unlike annotations, any whitespace on the front or back of the text will be stripped out when you compile. This allows you to insert "buffer space" around the footnote and help set it apart from the regular text. As with annotations, the placement of the bubble is important in that the text around it should flow sensibly, but with footnotes you also need to take into account that if the bubble were collapsed, that is where the footnote reference mark would be. Try to visualise the entire bubble as a single number, and this will help you with placement.

As a drafting application, Scrivener does not do typeset footnotes itself; rather, you define ranges of text that you want to be exported as footnotes. Thus, when the text in Scrivener is exported to RTF format, and opened in a compliant word processor, it will take on the familiar appearance of a numeral reference linked to a line at the bottom of the page with a matching numeral. Footnotes will also be exported in a variety of fashions for other formats as well, such as eBooks, MultiMarkdown syntax (and therefore as proper footnotes in LaTeX), as look-alike footnotes in HTML and plain-text, amongst others. For all options available, read Footnotes/Comments (section 24.21).

18.1.3 Referenced Inline Footnotes

If you would prefer to store the content of the footnote itself somewhere other than within the base text, see Linked Footnotes, in the following pages. Also note, if you require more than one note stream, you can couple linked footnotes with inline footnotes and treat them separately during compile.

If, however, you like inline footnotes but would still like to keep the bulk of the notes outside of the main text body area, there is a method you can employ which works in a cross-referencing manner. To use this method, place a footnote in the intended location within the text body, and type in a unique reference keyword surrounded by square brackets. An example of this might look like this: "[Reference]". Now anywhere else in the compile contents (usually the Draft folder), place the full content of the footnote and make sure to place that same bracketed reference in its content area. For the above example, this could look like: "[Reference] Here is the text of the footnote as it will appear in the final manuscript."

When compiled, Scrivener will match the two footnote ranges together by their bracket reference, remove the bracketed segment entirely, and shift the content so that it is defined in the location of the original anchor. Make sure the anchor and the text are both included in the compiled output, are both defined with an inline footnote range, and that each footnote has one unique reference pair.

Footnotes will prefer references that occur in the same document. So if for some reason you need "default" footnote text that is overridden in a local context, you could leave the

default in the external document, and supply an override in the same document that requires the special case footnote.

18.1.4 Finding Inline Notation

Scrivener provides tools for stepping through your entire project, searching for annotations and footnotes. Read more about this feature in Find by Format and Text Function (subsection 21.1.6).

18.2 Linked Notation

Creating and reviewing linked notation is made very simple with Scrivener's innovative sidebar approach. Unlike methods which use the margin areas beside the text to display the note, or typesetting driven methods which attempt to display a footnote exactly as it will appear in print, linked notes give you the power of placing meta text outside of the base text, while presenting this information in a stack which, while ordered as they appear in the document, does not otherwise reflect the spatial distance between notes. This means a footnote on page 12 will appear right alongside a footnote on page 27. This stacking method makes it easy for you to see all of your notes at once, and jump straight to that portion in the text, by clicking on these notes in the stack. When viewing a larger portion of the text with a Scrivenings session, will all be collectively stacked together in the sidebar in the order they appear in the draft.

Individual notes can be collapsed, so if one note is quite long, it needn't monopolise the space. Click on the disclosure arrow to collapse or expand a note, or use the Left and Right arrow keys on the keyboard (multiple notes can be selected at once). To collapse or expand all notes, use ⌘-0 and ⌘-9 respectively. Collapsed notes will show the first few words of the first line of the note. You can read the whole note by hovering over it with the mouse.

Meanwhile, the "anchor point" in the text where the note is attached will be drawn using one of three methods:

1. For comments, a prominent coloured highlight around the word, making it easy to spot the location of notes as you scroll through your text
2. For footnotes, a grey highlight around the word or phrase, giving a distinctive advantage over hunting for small superscript numbers in a word processor.
3. Alternatively, for footnotes, affixed after the word or phrase as a symbol or marker. This approach more closely mimics how a word processor presents footnotes and endnotes. It can be a useful alternative when the original text is densely covered with references.

Clicking on any anchor highlight will highlight that comment in the inspector, opening it if necessary to do so. You can also hover over a highlight, and the note text will appear in a tooltip without opening the Inspector.

There are three other places notes will be positioned:

1. In QuickReference panels, comments and footnotes will be placed into a split view below the main text editor within the panel.

2. In composition mode, comments and footnotes have a dedicate pane in the floating inspector.

3. Snapshots will save their notes when you take the snapshot, and you'll be able to see the anchors in the snapshot viewer, but to read the notes you will need to drag the snapshot to a header bar in one of the available editor splits.

To use linked notes you must in most cases have anchor text to attach the note to, in the same way you need some text to create a hyperlink with. Often this will be whatever text you wish to comment upon, or in the case of footnotes, the position where you want the footnote to ultimately appear. If you wish to jot down pre-writing notes in a blank document, it might be easier to use inline annotations or the document notes sidebar in the Inspector (section 19.4), instead.

To create a new note, either:

1. Select the text you wish to notate and then invoke the appropriate menu command or keyboard shortcut. To make a comment, use **Format ▸ Comment** or ⇧⌘–**8**. To make a footnote, use **Format ▸ Footnote** or ^⌘–**8**. This method is quite useful for comments. The highlight range doubles as the specifier; iIf you intend to address a piece of prose in your text, the highlighted text is immediately obvious as the problem area.

2. Place the caret in or after the word you wish to notate and use one of the above commands or shortcuts. This is the easiest method to use while writing, as you don't have to switch to selecting text. When you reach the point in the phrase where you wish to insert a footnote, you can merely finish typing in the word and press the shortcut key.

Some typesetting style guides prefer to place the footnote marker before the terminal punctuation mark of a sentence, especially if the footnote pertains to a clause within the sentence rather than the sentence as a whole. If you are writing in a language or style guide that calls for this, you can adjust whether the automatic consideration of a

"word" includes any following punctuation mark in the Formatting preference tab (subsection B.8.2), "Terminate footnotes and comments before punctuation". If you require a mix of styles, then you will need to use the first method of specific selection prior to making the footnote.

In addition to the menu and shortcut commands, you can also click either the **+** button or the **+fn** button in the header bar of the Comments & Footnotes inspector to add comments and footnotes to the selected text, respectively.

> **Getting Back to the Text**: Once you have finished typing in the content of the note, you can swiftly return to where you were typing by hitting the **Esc** key. In the case where you have pre-selected text to be notated, the cursor will be returned to the end of the prior selection.

Since linked notes are a type of link, they cannot be placed atop a range of text that is already linked, without removing the underlying link. The same holds true in vice versa: if you have a selection of text already linked to another document with a hyperlink or Scrivener Link, trying to annotate on top of that link will destroy the link, as text can only be linked to one destination at a time. To annotate a link, consider using inline notes, or place the link anchor around the linked text, rather than on it.

18.2.1 Managing Linked Notes

Deleting notes can be accomplished by selecting the note(s) in the inspector and then clicking the **-** button, by clicking the small **X** button in the top-right of each note, tapping the **delete** key or by simply deleting the text that the note is anchored to in the editor.

There are two ways to move a note from one text anchor to another:

1. Select the text in the main text editor that you wish to re-anchor the note to, and then right-click on the note itself in the inspector (not on the old anchor in the text, that will change your current selection). You will find a menu command to move the note's anchor to the selected text, **Move to Selection**
2. Simply drag and drop the note from the Inspector pane into the text where you wish it to appear. When using this method, the same logic will be applied as when creating a new note without first selecting a range. The nearest word (and its adjoining punctuation, if relevant) will be set as the new target. The exception to this is when using Footnote Markers instead of standard range-based footnotes, the marker will always be dropped precisely where you let go of the mouse button
3. In a slight variation of the above, if you pre-select text in the editor and drop the note on the selection, the selected range will be used to anchor the note.

These techniques can be a useful way of temporarily setting aside a note so you can edit a paragraph freely. When used in a scrivenings session, this technique also allows notes to be moved from document to document.

18.2.2 Linked Comments

Linked comments, as with inline annotations, can be assigned a custom colour. By default, when you use one of the previously mentioned techniques for adding a note, the highlighter anchor box in your text will be yellow, as will the corresponding note in the inspector. These two colours will always match, making it easier to see where you are in your text in relation with your notes.

To change the colour of a note, a convenience tool has been designed into the right-click contextual menu when clicking on the note in the inspector. Six default colours have been provided for rapid usage, but you can also opt to use the colour palette to change the colour to a custom selection. This can be done either by selecting the note and pressing ⇧⌘–C as per normal, or selecting **Show Colors** from the contextual menu. You can adjust the colour of many comments at once by selecting more than one before opening the colour palette, or right-clicking on any of the selected notes.

Scrivener will also remember the last colour you chose and automatically use it for the next new note that you create. This is remembered across all projects, and will be persistent until you specifically choose a new colour.

As with inline annotations, the primary usage for comments will be to add commentary to your text; commentary that isn't intended to appear in the final cut. However it is possible to adjust this behaviour in the compiler, and even opt to compile comments as footnotes. Read Footnotes/Comments (section 24.21) for further details on available options.

18.2.3 Linked Footnotes

As with inline footnotes, linked footnotes will by default appear with a grey background, and will likewise appear grey in the inspector. Footnote colours cannot be individually changed, though you can adjust the global application default for footnote presentation in the Appearance preference tab (subsection B.3.7).

Since footnotes will ultimately generate a marker in your text, the positioning of the anchor point in the text is important, as it will be used to place the reference mark in your final text once you compile. The positioning point is at the end of the anchor highlight area, not the beginning. If you wish to place a note at the end of a sentence, you would highlight up to the last word of that sentence (where you choose to start highlighting is at your discretion). This will be done for you automatically if you place the cursor anywhere in that word or at the end of the word, and create a footnote.

Footnote Highlight Style

There is an alternate form of usage which you may prefer to use if you find the default appearance too heavy for the quantity of footnotes in use. This method will place the footnote marker after the current selection, or current word, rather than highlighting them. What Scrivener uses for a marker can be customised, but by default it uses an asterisk. To enable this method of working, visit **Project ► Text Preferences...**, and enable the "Use footnote marker" option toward the bottom.

When compiling, the system will scan the contents of the footnote link anchor, and if contains this supplied marker and nothing else, it will remove the placeholder text from the compiled manuscript. Take care to select one marker and use it consistently from that point hence. If you switch markers halfway through your project, or turn this feature off, the mismatching markers will appear in your final draft. Marker text can be changed by hand in the editor, so if you need to switch styles, you'll need to settle the differences using the editing tools.

The **Make Default** button can be used to force this setting to all of your projects, even those that were not using it before, or had their own custom markers assigned. Again, make sure you have not already invested in a particular marker style before clicking this button.

See Inline Footnotes (subsection 18.1.2) for general tips on footnotes in Scrivener.

Compiled Footnote Numbering

By default, when you compile the document to any format, linked footnotes will be numbered in the order they appear, and this number will be saved into the footnote as a static identification of it. This is entirely unlike standard word processors, which generate the final document appearance while you are working. Scrivener's numbering system is designed to work in a more stable fashion, allowing author and editors to refer to static numbering that is tied to the last generated hard copy.

Consequently, the addition or removal of footnotes from the sidebar will not disrupt the stored numbering of existing footnotes. This way you can continue to write and make changes to the source project without losing frame of reference with the hard copy that is out for proofing. An option has been provided to prompt you whenever you compile. This way you can do test compiles on your own without disturbing the original numbering that is out for edit.

Compile footnote numbering can, with very large texts that have a lot of footnotes, take a long time to process. If you find compiling takes an inordinate amount of time, you should try disabling the automatic feature (causing it to work on-demand) via the **Format ► Options ► Prompt Before Updating Footnote Numbers** menu command.

18.2.4 Notation Text Scale

As with the main text editor, inspector notation can be scaled with a zoom feature in a way that does not disturb the underlying formatting, in one of two separate modes:

- Coupled with the editor zoom scale: if you have already zoomed the main editor text, it is likely you'd want to match text sizes in the notation sidebar. Simply right-click on any comment or footnote and select the **Zoom ▶ Use Editor Zoom** contextual menu command.
- In some cases this can produce undesirable results, such as when each editor split is using its own zoom setting. In these cases, setting the zoom to a fixed scale will uniformly adjust the text size. This value can be adjusted quickly with the ⇧⌘–, and ⇧⌘–. keystrokes.

To reset Inspector notation size to default, choose "100%" from the contextual pull-down menu.

18.2.5 Finding Linked Notations

Scrivener provides tools for stepping through your entire project, searching for inspector comments and footnotes. Read more about this feature in Find by Format and Text (section 21.1.6).

It is also possible to use the standard search tool for Document Find and Replace (subsection 21.1.1). With the Inspector open, searches will be highlighted in the sidebar for you, and replace will run in the background on the contents of your notes.

18.3 General Usage Tips for Notation

18.3.1 Stripping Out All Notation

If you wish to strip out all (inline and linked) notes from a text at once, this can be easily done by copying the text with **Edit ▶ Copy Special ▶ Copy Without Comments or Footnotes**, or ⇧⌥⌘–**C**, and then pasting directly over the copied text. Do note this will also destroy footnotes, so use this command with care.

18.3.2 Referring to Compiled Numbering

When producing compiled documents for your own proofing, or when coordinating proofing and editing with others, it can often be useful to be able to reference footnote and endnote numbering from within Scrivener. When you compile, all of your dynamic notes will be turned into the appropriate styling in word processors, including numbering. Scrivener will by default save these numerical assignments and display them in the Inspector pane.

These numbers will remain *static*. If you change the order of your outline, or add and remove footnotes, the numbers in the Inspector will not change dynamically as they do in a word processor. This way you can maintain a hard link with the currently distributed compilation, and when you receive notes from your peers regarding notes of such and such a number, can use Scrivener's Inspector pane to find those numbers, even if you've done extensive editing which would have otherwise altered those numbers.

Once you do compile, the numbers will be reset again. This might not always be desirable. To give a simple example, you might want to run off some personal compilations after distributing a proof, but if you've changed the order of things then your static numbering reference will no longer be in sync with the copy that is out for proof. In cases like this, you'll want to enable the **Format ► Options ► Prompt Before Updating Footnote Numbers** menu option. At the conclusion of the compilation, a pop-up warning will appear and ask if you'd like to have the new numbering pressed into the Inspector from this point on. If you click no, the numbers will remain in sync from the last time you clicked yes (or the last time you compiled without this option on, naturally).

If inline footnotes and linked footnotes are both set to output into the same stream (either both footnotes or endnotes), then the inline footnotes will be calculated into the numbering—but since there is nowhere to display meta-data for the footnote, the number of that inline footnote itself will not be displayed anywhere. The numbering of the linked footnotes around it will however jump over it, maintaining the accuracy of the count. Using this method, you can discern the number of an inline footnote by looking at the linked notes around it.

Additionally, with the RTF format you can also specify that linked comments also be exported as footnotes or endnotes. When this option is in play, numbering will take them into account as well.

For obvious reasons, there are a few caveats when using this feature:

- Notes must be compiled as sequential. If you are using settings which reset the numbering per page
- If you turn off numbering in the Inspector (via the **Format ► Options ► Hide Compile Footnote Numbers in Inspector**) then numbers will **not** be updated when you compile. This can be useful for freezing the current numbering without having to answer to pop-up message after every compile.
- For the purposes of recalculation, the compiler will consider *all* documents in the Draft (though it will only count those that are being exported). This way, if an item at one point had numbered references within it, and was later excluded, those numbers would be removed from the Inspector.

- When the compile group has been set to a Collection or to use the current Selection, if it also contains any items not in the Draft, then footnote renumbering will simply be skipped to avoid confusing situations. Ensure that your compilation Collections contain only Draft items, if you intend to use this feature on a regular basis.

- Numbering will always use Arabic numerals, even if the compiler was set to use another numbering scheme for the actual notes. If you intend to use this feature for referencing out to proofs, you might consider leaving your numbering set to Arabic during this portion of the process.

Not *all* documents in the project are considered. Use this to your advantage. If you want to make a hard copy of your reference numbering available into the indefinite future, you could create a duplicate of the entire Draft (or the portions you wish to preserve) into the main area of the Binder. Once there these numbers will never be removed by the compiler and so will always be visible and left as they had been when these files were last in use. Be wary of situations where you often compile off of a Saved Collection as this duplicate Draft might very well have items which match the Collection's criteria, triggering the case above where Collections of mixed Draft/non-Draft items will forgo re-numbering. If the Saved Collection is primarily intended for producing Draft compile groups, consider limiting its search scope to "Search Draft Only".

This could cause confusion in cases where these items are dragged in whole or in part back into the Draft, as duplicate numbers would be an issue until numbering is reset.

18.3.3 Resetting Linked Note Formatting

Both comments and footnotes allow formatted text within them. This means you can apply whatever formatting you wish into these, and they will be exported with this formatting when you compile (assuming your compile settings have not overridden footnote formatting). Consequently, adjusting the Formatting preferences (section B.8) for notes will not immediately impact them, as this might wipe out your formatting. You can however, just as with documents, reset the formatting to the application default by right-clicking on selected notes and choosing **Convert to Default Formatting**. This menu option will only be available when the note is out of editing mode.

18.3.4 Exporting Annotations and Comments

You can export all of the inline annotations and linked comments in your project into an RTF file with **File ► Export ► Comments \& Annotations**. Optionally, you can restrict this to the selected items in the sidebar, rather than the entire project. You can also choose to export titles along with the comments, making it easier to find where they came from.

18.3.5 Notation with Copy and Paste

Within Scrivener, you can copy or cut, and paste ranges of text that contain any style of notation, and it will be carried along with the selection. This can even be done in between projects.

When pasting into other programs, however, the footnotes and comments will be converted to several styles that the receiving program can choose from. For word processors, most will select RTF formatting. Thus, to get the most benefit from pasting, you should use applications that can read RTF formatting. A good rule of thumb is, if it can import footnotes and comments, they will work with copy and paste, too.

Pasting into many other applications will produce a different result, as most do not support either footnotes or comments. In this case they will use RTFD formatting, which is a "flattened" version that looks like footnotes and inline commentary, but in effect just formatted text. Since these applications cannot use real footnotes and comments in the first place, this is an acceptable compromise.

> **Pasting Into Word 2011**: Microsoft has changed how Word accepts pasted text to prioritise the use of the RTFD method instead of RTF, even though it can handle RTF just fine. This means you'll end up with flattened notation as though Word cannot support them. If you use Word 2011, you'll want to enable the "Use Word-2011 compatible copy" in the Import & Export preference tab (subsection B.10.3). Do note that when this option is enabled, it will cause standard applications which would ordinarily use RTFD to fail, as that flattened version will no longer be provided. So only enable this option if Word is your primary processor.

Finally, pasting into a plain-text editor, like TextMate, BBEdit, or TextEdit in plain-text mode, will produce a plain-text flattened alternative, which uses brackets and line-spacing to achieve a footnote and comment effect.

If you wish to copy and paste text *without* any notation at all, use **Edit ▸ Copy Special ▸ Copy Without Comments and Footnotes**, or ⇧⌥⌘–C.

18.3.6 Links, Notes, and Tables

Inline notation supports the ability to create Scrivener Links within them. They will function as per normal, and this is a handy way to "hide" link text from the final manuscript. Owing to how linked notes work, it is not possible to create links within them. Consequently, if you convert your inline notes to linked notes, you will lose any Scrivener Links that had been set within them.

18.4 Text Colour and Highlights

Arbitrary text colour can be applied to your document in a fashion similar to any other formatting range. Select the text you wish to change the colour on, and either use **Format ▶ Font ▶ Show Colors** (⇧⌘–C), or use the text colour selector in the Format Bar (subsection 15.4.2) to set the colour. The format bar will remember the last colour you have used, so it easy to rapidly apply the same colour to multiple text selections with a single click. It will also pick up a colour when you click on text that has already been coloured. If you wish to change the colour, right-click on the colour chip or click and hold. A pop-over will appear with the following contents:

1. The built-in presets (black, red, etc.)
2. Common swatches (the first swatch with a red line drawn through is "remove colour" choosing this one will erase any colour in the selection and return the text to default)
3. If applicable, any custom swatches will appear in a separate grid below the main grid
4. Show Colors (access to the colour palette)

Custom swatches are imported from your colour palette in the custom colour shelf, which can be seen here (Figure 18.1). These are named automatically by the Mac OS X system. You may find similar colours will acquire the same name, this is normal; the names should be considered conveniences, not identifiers.

Setting a highlight to a range of text is quite similar, only you will use the second colour chip on the format bar, instead. As with the text colour control, single-clicking once will re-use the last highlighter colour, and holding the button down or right-clicking will access the full menu. For highlighters, this can also be accessed in the **Format ▶ Highlight ▶** sub-menu. In addition, the last colour used will be assigned to the keyboard shortcut, ⇧⌘–H for rapid application at any time. Scrivener will remember the last highlighter colour used from that point on, even in other projects.

The contents of the highlight menu will be divided into four main sections:

1. Remove Color
2. Built-in presets (black, red, etc.)
3. Custom swatches
4. Show Colors (accesses the colour palette so you can select a custom colour)

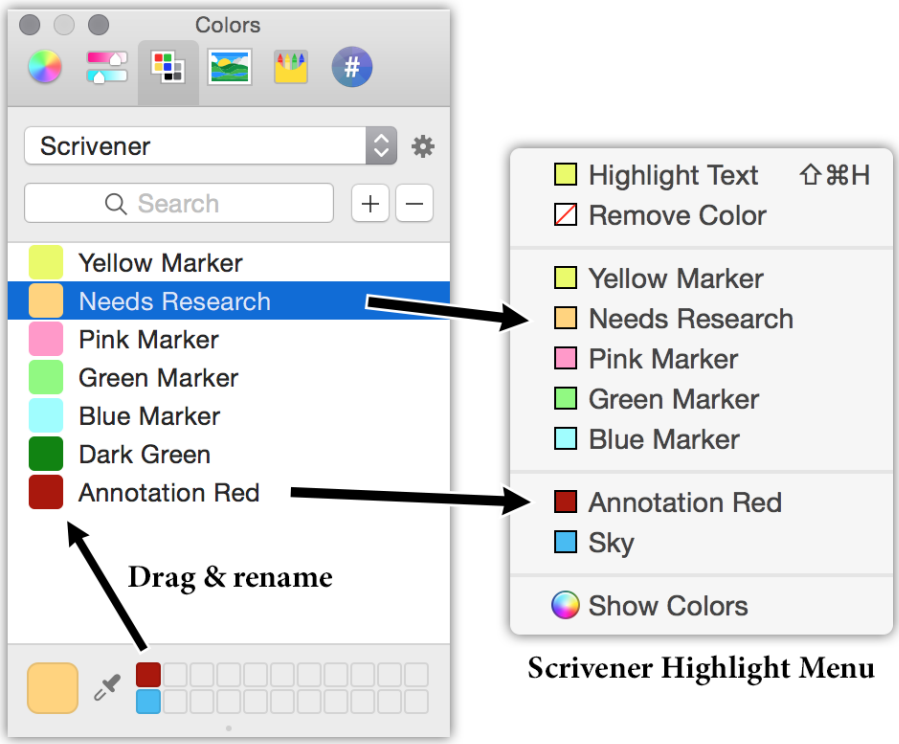

Figure 18.1: Scrivener swatch set

> **Colours and Compiling**: By default, text colours and highlights will be compiled into your final manuscript. If you use these features for internal editing, you will find options for disabling them in the Transformations compile pane (section 24.15). Not all word processors support text colour and highlighting features.

Scrivener provides tools for stepping through your entire project, searching for text colour and highlighters. Read more about this feature in Find by Format and Text (section 21.1.6).

18.4.1 Naming Text Highlights

The default text highlight marker names can be changed to something less generic, on a universal level, by using colour palette swatch groups. Swatch groups are OS X's way of letting you create selections of colours and giving them useful names. To change the highlighter colours, show the Color Palette (⇧⌘-C), and select the third icon from the left.

Next, from the "Palettes" drop-down menu, select the group titled, "Scrivener". You should now see something resembling the above screenshot.

> **Mac App Store Note**: It is not possible to install the custom Scrivener colour swatch group using the version of Scrivener for the Mac App Store. The Scrivener group will still show up in the palette, but any changes you make to it will be lost after you restart the program. To make permanent changes, you'll need to create your own group called "Scrivener Custom". The best way to do this is to rename the Scrivener set to "Scrivener Custom", using the gear menu to the right of the palette selection menu. This will create a persistent colour swatch group for you, preloaded with all of Scrivener's built-in colours.

Double-click on any of the provided highlights to rename it. Your changes should be visible immediately in the Scrivener interface, where applicable.

Your own colours can be added to any of Scrivener's various colour selection menus, as well. If necessary, click and drag downward in the location marked above, "Custom colour shelf". You should see a row of white squares appear. You can drag out as many rows as you like.

The easiest way to create new custom colours is to use either the colour wheel (first icon) or the sliders (second icon). When you find a hue that you'd like to use as a highlight, text colour, or annotation colour, drag from the large colour preview area at the top of the pane, and drop into one of the empty slots in the custom shelf.

If you check one of Scrivener's colour menus right now, you'll see that it has already been added below the base sets. To change the name of your custom colour, click on the swatches icon again, make sure the Scrivener set is active, and drag the colour from the custom shelf into the list. Now that it is there, you can double-click to rename it.

Note that due to the way colours are estimated, if you have several subtle variations they will all get the same name. To avoid this, create a new named colour swatch for each variation. This is a purely aesthetic feature though. Names are not used for anything, except for your own reference.

> **Mac OS X Tip**:If you wish to share your colour sets with other users, or transfer them to another computer, you will find a file in your user Library folder, under Colors, named Scrivener.clr. Place this file into the same folder on the second computer and it should become instantly available.

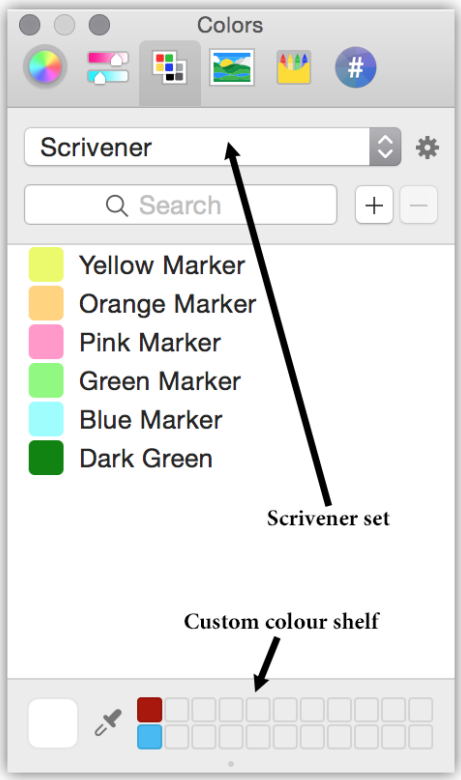

Figure 18.2: Custom named colours

18.5 Marking Revisions

Scrivener uses a basic technique for making revisions visibly distinctive from one another and the base text. At its core, the feature is simply a formalised method for apply text colour as you type. The formal nature of this is in both the selection of the tool, rather than picking some arbitrary colour from the Format Bar, and a collection of tools for working with that particular colour at a later time.

Whenever a revision level has been enabled in a project, it will impact all primary editor views where text editing is possible (it will not impact notes or other ancillary fields). The cursor colour will be changed to that colour, to remind you of this fact, before you start typing. In addition to colour the text as you type, usage of the strikethrough format will colour the strike-out in accordance with the revision level, too.

> **Limitations**: Revision levels are not to be considered equivalent to "track changes", as seen in version controlled software, or Microsoft Word. They are very simple, low bandwidth tools, that only mark modified or new text. Removed text will not be marked in any way. If you wish to track all changes between a document and its

revised state, take a look at the snapshot feature (section 15.6). Stricken text, while it will display the revision level from when strikeout was used, will need to be manually removed later on.

18.5.1 Setting a Revision Level

There are five revision levels available, and each are accessible in the **Format ► Revision Mode ►** sub-menu. Simply click on the coloured menu command you wish to use as your revision mode, and begin editing the document. As you type in new text, it will be automatically coloured with the revision level, as if you had typed it in and then selected it, and applied a text colour with the Format Bar. If you wish to paste text into the editor using the revision colour, use **Edit ► Paste and Match Style**.

Selecting the same colour from this menu again will toggle the revision marking system off. This can be especially useful if you bind these colours to keyboard shortcuts, using your Mac OS X custom keyboard shortcuts. If, for example, you bind Ctrl-2 to level two, then you can tap that shortcut again to return to standard editing.

18.5.2 Marking Existing Text

There are times when you may wish to mark existing text with a revision level. To do so you must already have a revision level set, then select the text you wish to mark, and use the **Format ► Revision Mode ► Mark Revised** menu command. This menu command will only appear when active text has been selected, so if you do not see it, ensure that the text you have selected is in the active editor.

18.5.3 Removing a Revision Level

Once you have reviewed a document's revisions, you will probably wish to remove the revision markings from that document. This can be very easily done by selecting the appropriate revision level from the menu, and then without selecting any text, use the **Format ► Revision Mode ► Remove Current Revision Color** menu command. This command will work on the entire text session that is in the active editor, so if you are editing multiple documents in Scrivenings, ensure you wish to wipe out markings for all visible documents. If you only wish to strip out marks from a portion of the session, use **⌘–4** to isolate the document, first.

You *can* also use this command in conjunction with a selection, to remove only a portion of the revision markings from the document. Make sure that the selection only contains text marked with that revision, and use the same menu command above.

Since revision levels are nothing more than coloured text, if you happen to have other text in your document that is coloured identically, this command will strip out those

colours too. In general, it is a good idea to reserve these colours for revisions only, if you intend to use the feature heavily.

18.5.4 Removing all Revision Markings

When you've reviewed all revisions are are ready to return the document to a default colour state, you can use **Format ► Revision Mode ► Remove All Revisions** to strip out all revision level markings in the active text session. The same warning from above applies: this will impact *all* documents in a Scrivenings session. Likewise, the same warning about overlapping colour usage applies as well: any colours matching any of the revision levels will be stripped from the documents.

18.5.5 Changing the Revision Colours

It is possible to change Scrivener's associations for which colours are which levels. These can be configured in the Appearance preference tab (subsection B.3.7), in the Revision Colors tree of the Customize Colors section.

Once changed, Scrivener will no longer associate existing coloured text as being a part of a revision, even if a revision mode was used to apply that text in the past. In particular, the tools for stripping out revision colours will cease to work on documents marked up with the original colour scheme, and the tool for finding revision markings (below) will no longer uncover these markings. It is thus advisable to not touch these settings if you intend to use revision markings in a collaborative environment. If multiple authors are using different preferences, none of the special tools for working with these modes will function correctly. Either establish a standard set of colours between all involved parties, or agree to use the defaults.

18.5.6 Finding Revision Markings

Scrivener provides tools for stepping through your entire project, searching for revision levels. Read more about this feature in Find by Format and Text (section 21.1.6).

18.6 Text Markings and MultiMarkdown

Since MultiMarkdown has scant tools for working with text formatting, many of the above marking techniques have no direct analogue, with the exception of footnotes. In the case of annotations and comments, it is possible to use embedded HTML coding to simulate comments, and they will be marked in such a way that it is easy to get both the colour information and the fact that they are comments, from an XSLT stylesheet. Other markings, such as text colour, highlighting, and revision levels, will be completely dropped from the compile.

18.6.1 Using Comments with MultiMarkdown

Inline annotations and inspector comments work very similarly when compiling with MultiMarkdown. Both can be independently stripped from the document during compile, or left in, making it possible to leave one as "functional" and the other style as pure commentary.

Since there is no direct analogue for comments[1], Scrivener employs the usage of MMD's ability to handle raw HTML coding in the document. When using MMD, it is possible to pass HTML through the parsing engine so that it lands directly in an HTML document, in a transitive state toward another format, when using XSLT post-processing, or as a final product.

This behaviour can be disabled in the compile settings, and comments will be passed through to the document in the same manner that they are using the plain-text export, enclosed in markers.

For advanced usage tips on using comments and annotations in a functional capacity, as a way of dynamically expanding MMD's base syntax, read Functional Annotations and Comments (section 27.1).

18.6.2 Using Footnotes with MultiMarkdown

When compiling a draft using MultiMarkdown, the compiler can be instructed to produce proper footnote syntax for all of your inline and linked footnotes in the project. This will be done using MMD's referencing syntax, where a footnote reference is placed at the point of the note in the text, and the content of that note is placed at the bottom of the file with a link to that reference marker. Scrivener will generate all of the reference tokens so that you do not have to worry about coming up with unique markers. Just use Scrivener's footnote tools, compile, and you'll have a completed document with footnotes.

If you do not wish to see footnotes in the final product, you can disable this function in the compiler, under the Footnotes/Comments pane, by checking "Remove footnotes".

For advanced usage of both inline and inspector footnotes as multiple note streams, read Multiple Footnote Streams with MMD (section 27.2).

[1] While it is true, MMD uses HTML, and HTML has a perfectly valid syntax for comments, it is not viable to use it because MMD already has an advanced secondary usage for HTML comments, in that raw LaTeX can be passed directly through the parsing engine and land in the final .tex file. Thus, usage of the HTML comment engine for general comments or other purposes would cause undesirable side-effects.

18.6.3 Text and Highlight Colour with MultiMarkdown

In short, nothing is done to export Scrivener's text colour (including revision levels, which are just a formalised way of using text colour) and highlighting features to MultiMarkdown, as there is no suitable analogue for these tools, and in most cases since they represent a function of a document's meta commentary, are not considered to be suitable for *semantic* transfer. If there are cases where you would like to use colour in your compiled documents, you might wish to take a look at advanced inline annotations, above.

Rather than looking at this as a limitation, you can consider it to be a feature. As with most purely rich text features in Scrivener, they can be used liberally to annotate and comment upon your work, without fear of them every showing up in the final product.

Chapter 19

Inspector

The Inspector is panel on the right-hand side of the project window that displays all meta-data, snapshots, and notes associated with the document shown in the editor that most recently had the keyboard focus, or in the case of a corkboard or outliner, the last card or row selected (which implies the container itself when all cards/rows are deselected). Since the special root folders in the binder—Draft, Research and Trash—have no meta-data associated with them, the Inspector just shows the project notes or references when these items are selected.

You may toggle the visibility of the Inspector with **View ▶ Layout ▶ Show (Hide) Inspector**, or by pressing ⌥⌘–I.

The Inspector is broken down into sections, or "panes". These panes can be selected by clicking the buttons along the top of the Inspector (Figure 19.1). The first four display the common meta-data features of the item, while the snapshot and linked notes section has dedicated panes that take up the entire Inspector.

With the exception of the aforementioned root folders, any item in Scrivener has access to all meta-data and notation capability. You can use Notes to write about a painting you are studying in the editor, attach gallery URLs and research papers to it with References, or organise it with keywords. Snapshots and linked notes however will only be available to items which allow text editing in the editor, namely folders and files.

19.1 Synopsis Card

At the top of the inspector (for the Notes, References, Custom Meta-data and Keywords panes), just beneath the buttons to switch Inspector panes, you will find a depiction of the document's index card. (The corkboard's special features, such as pins and stamps,

Figure 19.1: Inspector panel buttons, with "Keywords" selected.

will not be rendered on the card, as they can be readily viewed in expanded fashion in the inspector) As with the corkboard, you can edit the title and synopsis right on the card, updating the document's corresponding information.

There are two controls along the top of the card on the right-hand side. The first is a drop-down menu with an index card graphic in it. Click here to select between text synopsis and image synopsis. In graphical mode, you can drop in an image from your desktop (or the binder) which will be used to "represent" this document on the corkboard. The text synopsis will not be erased, and in fact will still be used in the outliner and various export methods, but in the inspector and corkboard, the graphic will be used instead. Read Placing Images on the Corkboard (subsection 12.1.7) for further information.

The Auto-Generate button, on the far right, will produce a synopsis based on the first few lines of text in the document. If text is selected in the editor, it will be used to fill in the synopsis; otherwise, the first few lines will be used. Clicking this button is similar to the **Documents ► Auto-Generate Synopsis** menu command, however the latter does not consider your current selection. When using image synopsis mode (subsection 12.1.2), this button will be replaced with an **X** which when clicked will erase your graphic selection.

The Synopsis section can be collapsed to make space for the other elements of the inspector.

See also: So What are Index Cards, Anyway? (subsection 12.1.1)

19.2 General Meta-Data View

This collapsible section appears in the middle portion of the Notes, References and Keywords panes of the Inspector. It displays the basic document meta-data and provides controls for editing it. Read Meta-Data Types (section 10.1) for further detail.

19.3 Document Support Panes

The six views of the inspector present further access to notes, reference links, keywords, custom meta-data, text snapshots, and linked comments and footnotes, respectively. The first four share space with the synopsis and general meta-data panels, the last two take up the entire inspector interface.

The six panes can be accessed at any time by clicking on the icons at the top of the inspector, or by using the **View ► Inspect ►** sub-menu and its respective shortcuts.

19.4 Document and Project Notes

Document notes provide a rich text writing area that is outside of the main text area. You can use notes to store information about the current document for reference whilst writing, without cluttering up the binder with note files or the main text with comments and annotations. Clicking the header title of the notes area will bring up a pop-up menu where you can select "Document Notes" or "Project Notes". Document notes are always attached to the current document in the sense that you cannot view them unless you have loaded their respective document in the Inspector.

Project notes are similar to document notes, except they are visible to the entire project, no matter what document you have selected. In addition, you can also create multiple project notepads, using the Project Notes window (section 12.3), or by selecting the "Manage Project Notes…" menu command at the bottom of the Notes header bar selection menu. When more than one notepad has been created for a project, you will see them appear in this menu as well. There will always be at least one project notepad.

You can cycle forward and backward through all available document and project notepads with ⌘-6 and ^⌘-6.

19.5 References

The References table allows you to hold links to related material within the project itself, elsewhere on your hard drive or on the Internet.

General, or vague linking between items can be a useful thing to do. Linking specifically from a piece of text, using the hyperlink format, certainly has its place, but sometimes an item only generally applies to a large group of words. Fortunately Scrivener has a tool for this called References. Consider if a sub-section in your book has several supporting articles that you've imported into the Research folder. You reference these frequently when writing in this section of the book, but it's a pain to always have to look them up. Dropping these articles into the References inspector pane establishes a *general* link to them pertaining to the whole section of the draft—that document.

Another thing they accomplish is to establish a low-impact network of back-links. Whenever you create a Reference or Scrivener Link in your document, the software adds a Reference link in the item you linked *to* back to the item you linked *from*. Thus, if subsection 12.23 links to a paper on Colony Collapse Disorder, then the next time you are reviewing that resource, you could open up the Reference pane and see all of the sections of your draft that have established a link to it, section 12.23 included. You might prefer that Scrivener never do this, and curate all links yourself. If so, disable this feature in the Navigation preferences tab (subsection B.6.7).

To create links, simply drag documents from the binder or the Finder, and URLs

from your browser, into the table to create links. You can also use the + button in the References header bar.

Double-clicking on the icon of the reference, or selecting it and hitting the Return key, will open the document, either inside Scrivener or in the default application or browser depending on the reference type. How internal (and supported) file types are opened in the editor is determined in the Navigation preference tab (section B.6). This sets the default, but you can opt to open a reference using an alternate method by right-clicking on it in the table. It is possible to open references in a:

- Current View, replacing the current editor
- Other View, as a split view using the last split type you used if one needs to be created
- QuickReference panel
- Quick Look, which will only be available for unsupported file types

Using these menu commands, or dragging and dropping reference items to an editor header bar, will let you open external media files not stored in the project, in the editor as though it were a project resource.

The + button has three options available:

- *Add Internal Reference*: presents a menu containing all of the items in your project binder. Click on any item, even a folder, to create a reference to it. The default description for it will be its binder title.
- *Look Up & Add External Reference*: use this menu to load a file browser. Any file you select in this browser will be added as a link to the reference list. The default description for it will be the file name.
- *Create External Reference*: creates a new row in which you can manually copy and paste or type in the internal title and valid URL or file path of the resource you wish to link to. This is most useful for creating links to the Internet, as you can just paste in the URL you have available, but local file URIs are also valid.

Any existing reference can be edited by double clicking on the text labels. To finish editing, press Return or click anywhere else in the table with your mouse. You can use this to update broken links with the correct file path or URL, or change their internal names. For internal links to other Scrivener items in your project, you cannot edit the

target, only the name. To replace it with a new target, you'll need to create a new reference and then remove the old one.

Clicking the header bar of the references area will bring up a pop-up menu from which you can select "Document References" or "Project References". As with notes, document references are attached to the current document whereas project notes are global and remain the same no matter which document you are viewing.

To delete a reference, select it and click the - button, or tap ⌘–Delete.

[**10.7+ (Lion) Only**] References to files on your disk can be dragged into a header bar for viewing, even without importing them into the project. If Scrivener does not know how to display it, it will present the file's Quick Look information. In versions of the operating system older than 10.7, URLs and unsupported file formats will not allow a drop on the header bar.

External references can be dragged out of the table and into the Binder. This will result in the resource being *imported* into the project if it is an external reference; this goes for web pages as well.

When you create a Scrivener Link to a document in any way, a Reference back to the originating document will be placed in its references list. This way you can easily track "back links" while browsing your project. These links are, however, not dynamic. They are created once upon linking, and can later be removed or altered by hand.

It is now possible to drag items from the binder of one project into the References pane of another (either project or document references are a valid target). This will store a special external link to both the project it came from, and the individual item you dragged into the References table. Double-clicking on the icon will load the project if necessary and reveal the target item in the active split.

> **Pro Tip**: If you change the "URL" for internal links (which otherwise just state "[Internal Link]" in the URL field) to a single asterisk, that item will be placed at the very top of the header bar icon menu. If accessed from there, it will load that file in accordance with your navigation settings for internal references. Use this to create a more visible "soft link" between one or more documents, such as frequently used research material, instructions for preparation, and so on.

19.6 Keywords Pane

Keywords can be used to hold arbitrary information. A few examples might be character names, locations, interviewees, or workflow hints such as "Needs Illustrations". You could assign the name of a theme to a document as a keyword, and then later you could

perform a keyword search to find all documents that deal with that theme (especially useful as you are unlikely ever to use the name of a theme directly in the text itself).

You can associate keywords with documents to help you find them when performing a project search. Use the plus and minus buttons in the header bar above the keywords table to add and remove keywords (or you can drag words and phrases from text to the keywords table to create a new keyword). You can also use the Project Keywords window, which stores a repository of all keywords used in the project, to assign keywords by dragging them from the project keywords into the keywords table in the inspector; this is an easy way to add several keywords at once.

There are several ways to add new keywords to your document:

1. Drag the keywords from the Keywords panel to keywords table in the inspector
2. Drag the keywords from the Keywords panel to the document header bar, or its name in the binder (this latter use allows you assign keywords to many selected documents at once)
3. Click the **+** button in the keywords table header bar to type in a new keyword. If the keyword already exists, auto-completion will be used to help you with data entry. You can also use the **Enter** key in the table to add new keywords
4. Using the gear menu in the keywords table header bar to access the **Add Keyword** menu, which will contain a sorted list of all keywords in the Project Keywords panel
5. Right-click anywhere in the keywords table itself for quick access to this menu, as well.

Keywords can be unassigned from a document by selecting them in the keywords table and clicking the **-** button, or pressing the **Delete** key on your keyboard.

To change the order of the keywords, drag and drop them in this table. This can be useful if you are viewing keywords on the corkboard as colours. Since only the first few (the precise number of which you can modify) will be shown, it is good to move the most important keywords you intend to track to the top of the list.

Viewing a particular keyword in the panel is easy to do, just double-click on the keyword's colour chip that you wish to see and the Project Keyword window will be opened for you and will select the keyword in the list.

See Also:

1. Using Keywords (subsection 10.1.4)

2. Project Keywords (section 10.3)

19.7 Custom Meta-Data Pane

Custom meta-data is used in two primary locations: the Inspector panel which corresponds with the fourth button from the left, which looks like a suitcase tag; and the Outliner, which can display individual meta-data fields in a column. Read more about the overall philosophy and usage of Custom Meta-Data (subsection 10.1.6).

In most new projects, there will be no fields set up, and this pane will merely display a message to that effect, including a button which will take you to the set-up window, documented in the prior cross-reference. This can also be reached at any time via the gear menu, under "Edit Custom Meta-Data Settings…", or via the **Project ▸ Meta-Data Settings...** menu command.

The pane represents each field as a titled plain-text block that you can freely type within. Each field will be stacked one atop the other in the order that they appear in the project settings table. You can use **Tab** and **⇧Tab** to move between fields. The currently active field will be highlighted in blue.

Press the **Return** key at any time to conclude editing the field (though clicking anywhere else in the interface will also suffice). If the pane has been focussed with the hotkey (**^⌥⌘–M**) then tap the UpArrow or DownArrow key to select the first or last field respectively.

The gear menu presents two display options that pertain to this pane:

Word-Wrap All Fields Off by default, meaning that fields which have not been set to word-wrap will only ever show one line at a time in the display. If you would prefer to see all of the text in each field, regardless of its configuration, enable this option.

Color Field Names On by default, the row colour (if relevant) for the field will be used to print the title. Disable this if the colour settings you have used make this pane difficult to read.

19.8 Snapshots Pane

This section of the Inspector gives you easy access to all of your existing snapshots, the ability to create new ones, load up comparative displays between snapshots or the current editor, and delete old ones.

At the top of the Inspector, you will find two buttons. Clicking the **+** button will create a new untitled snapshot for the current document (you can also use **⌘–5** to do this

while typing). To remove a snapshot, select it in the table, and click the - button. To read the contents of a snapshot, click on any of the rows and its text will appear in the text preview area below the table. It is not possible to edit snapshots, but you can select text from them and use copy and paste or drag and drop to restore bits of them.

The **Roll Back** button will revert the current editor text to the text in the selected snapshot. To use it, select the snapshot you wish to revert to, and click the **Roll Back** button. Scrivener will request confirmation from you, and give you the opportunity to snapshot the *current* text if you wish. Note this cannot be undone, but if you opt to create a snapshot from your current text, it is possible to effectively undo a roll-back by rolling "up" to the latest revision.

The comparison control features let you quickly see differences between the snapshot and the current version of the text, or between two snapshots, by selecting more than one snapshot from the table. When in comparison mode, the button title will change to **Original** which will return you to the non-annotated preview when clicked.

The two arrow buttons which will appear in comparison mode let you jump from change to change. You can use ^⌘-(and ^⌘-) to navigate between changes, as well.

In some cases, the level of editing will be as such that the comparison mode produces confusing results. The method of analysis can be fine-tuned by clicking the down-arrow beside the comparison button at any time. In many cases, removing word-level analysis will produce a cleaner result, when extensive edits have been made. These options work in a descending fashion, meaning that the lowest selected option overrides the options above it. Thus, to switch to "By Clause", you need only disable "By Word".

1. *By Paragraph*: any changes made within a paragraph will trigger the entire paragraph as having been modified.
2. *By Clause*: any changes made within a clause (as in a sentence) will mark the entire clause as having been modified.
3. *By Word*: individual words will be marked, producing the most precise (and thus "noisy") results.

See also: Using Snapshots (section 15.6).

19.9 Linked Notes Pane

The final pane keeps track of all the comments and footnotes that appear in the current text view. This means that in Scrivenings mode where multiple documents are represented, this display will show a combined view of *all* notes across the documents, in the order that they appear.

Along the top of the pane you will find three buttons that, in order of appearance from left to right, do the following:

- Add a new comment in the text at the current text position or selection
- Add a new footnote in the same manner as above
- Delete the selected comment(s) for footnote(s).

Select multiple items in the same fashion you would select multiple documents with the Cmd and Shift keys. Notes can be independently collapsed or revealed by clicking the triangles beside their title. The ⌘-0 and ⌘-9 universal shortcuts for collapse all and expand all work as well here, but you may also Opt-click on any arrow to toggle them all in the same direction as the arrow you clicked on.

Press the Esc key to return to the main text editor at any time, even while editing a note.

> **Using notes as bookmarks**: Clicking on a note will scroll the text editor to the position that they appear in within the text. This means you can use notes as a form of bookmark, or quickly jump through your text point-by-point to areas that need addressing.

Right-click on any of the notes to convert between footnote and comment, or to reset their formatting to the application default, which can be set up in the Formatting preference tab (subsection B.8.2). Additionally, comments can have their colour changed by right-clicking on selected comments. This will also change the anchor highlight in the main text.

See also: Linked Notation (section 18.2) for full usage notes on the comments and footnotes themselves.

19.10 Locking the Inspector

When using two editor splits, you can lock the Inspector to the current editor so that when the other editor in the split view receives focus, the Inspector does not change contents but instead continues to display information about the document in the editor you locked it to. Click on the icon of the padlock in the top right corner of the Inspector to toggle this state.

19.11 Advanced Inspector Keyboard Usage

The following keyboard shortcuts may be used to access the individual Inspector panes:

- ^⌥⌘-H — Notes
- ^⌥⌘-N — References
- ^⌥⌘-J — Keywords
- ^⌥⌘-M — Custom Meta-data
- ^⌥⌘-K — Snapshots
- ^⌥⌘-L — Comments and Footnotes

These five keys are all placed under the right hand in ascending order, making it easy to memorise them and use them without leaving the home-row. These shortcuts have two modes of operation. If the pane in question is not visible, the Inspector will open (if necessary) to that pane. If the keyboard shortcut is pressed while the particular pane is already visible, it will be *focussed*. This means you can always start typing in notes, even if the Inspector is hidden, by quickly tapping the corresponding keyboard shortcut twice. The focus model is used in the panes like so:

1. Notes: Immediate access to existing text so you can start writing and editing in notes immediately. Switch between notepads with ⌘-6 to select the next notepad in the list, and ^⌘-6 to select the previous notepad in the list. These shortcuts will "wrap around" the list so you can continue to use one of them to cycle through all notepads.

2. References: Selection is moved to the list so you can navigate with arrow keys; tap Enter to activate a link; or Del to remove a link from the list.

3. Keywords: Selection is again moved to the list. Use arrow keys to navigate between existing keywords; tap Enter to add new keywords; tap Del to remove keywords.

4. Custom Meta-Data: Nothing will be actively selected, but you may use the arrow keys to move between the editable fields.

5. Snapshots: Focus is moved to the text preview field so you can quickly copy and paste from here if necessary.

6. Comments & Footnotes: Focus is moved to the note list. Use the up and down arrow keys to move between notes and jump to their corresponding position in the main document; **Enter** to edit the text of an existing comment or footnote; left and right arrows to collapse or reveal a note; and **Del** to remove notes.

These shortcuts can be used in the following places as well:

- QuickReference panels: if the panel has the focus, then using one of these shortcuts will open the split view to the inspector info requested by the shortcut
- Composition mode: likewise, when in composition view, these shortcuts will call up and reveal the appropriate portion of the floating inspector.

Chapter 20

Scriptwriting

Although Scrivener is not intended to be a dedicated scriptwriting program (for such a program you might want to try Final Draft or Celtx if you have not done so already), it does allow for basic script formatting and export to formats that most scriptwriting software can read, making Scrivener suitable for first drafts.

> **Default Scriptwriting Font**: By default Scrivener will use the Courier Prime[1] font for all scriptwriting modes that would make use of Courier as a font. Courier Prime is specifically designed for scriptwriting, using the same font metrics as Final Draft Courier and standard Courier, meaning you can rely upon using it for accurate page counts. If you do not like Courier Prime, or would like to use Final Draft Courier if you have it installed with Final Draft, then you can create your own scriptwriting preset (section 20.7).

20.1 Formatting a Script in Scrivener

To format a script in Scrivener, select the format you want to use from the **Format ► Scriptwriting ►** sub-menu, or use choose one of the pre-built project templates in the "Scriptwriting" category. When in scriptwriting mode, the top item in the Scriptwriting sub-menu will be checked and will display the name of the format you are using. This preferred format can be toggled on individual documents with **⌘–8**.

By default, the standard "Screenplay" format is selected and Scrivener is in scriptwriting mode. If "Script Mode - Screenplay" did not have the tick next to it, then we would know that the current editor was not in scriptwriting mode (that is, it would be in normal prose mode for general writing). You can more easily tell whether or not you are in scriptwriting mode by looking at the footer bar, which will display various scriptwriting tools and hints, rather than the standard word and character statistics display. Additionally, the binder icon for that item will be tinted yellow, with three-hole punches on the side.

Scriptwriting mode is a setting which is individual to each document. Once a document has been toggled to scriptwriting mode, it will remain that way until you change it.

Thus it is possible to have a standard document in one split for your notes, while using scriptwriting mode in the second split to draft your work. However, it is not possible to use more than one *type* of script formatting in the same project. This is a project level setting, and so if you need to create a stage play and screenplay at the same time, you will need to do the adaption in a tandem project.

At the bottom of the scriptwriting menu you can see a list of all the different script formats that are available. Scrivener comes with the following formats built-in:

- Screenplay
- BBC Radio Scene Style
- BBC Taped Drama
- Comic Book (Antony Johnston)
- Comic Book (Alternative)
- Stage Play (UK)
- Stage Play (US)

You can, of course, mix up different text modes in the same document or draft, so that parts of a document may be written as a script and other parts written as general text. This makes it very easy to write treatments in Scrivener, or to intersperse general notes with script.

The screenplay format that comes with Scrivener by default is based on a Hollywood standard, but with half an inch added to the left indent of each element so that when you print with a standard set-up of one inch margins, the left margin will actually start at one-and-a-half inches, which is the standard to allow for three-hole binding. In other words the end result will be accurate, even if the numbering by the editor's ruler is slightly off.

The scriptwriting mode in Scrivener works much like other scriptwriting programs (primarily, Final Draft). On the right end of the footer bar is a drop-down menu containing the various elements for the selected format. Clicking on an element will reformat the current paragraph to the format of the selected element. So, for instance, if you clicked on "Character", the current paragraph might all be capitalised and centred.

This menu also has full keyboard access, allowing you to change modes swiftly and without the use of the mouse. The menu can be called up with ⇧⌘-Y at any time from within a scriptwriting document (even in composition mode). To select an individual element style, tap the associated letter for it, on the right side of the menu. Example: "T" for Transition, and "C" for Character. Additionally, each element (up to the ninth) will

be assigned with a number (in the order that they appear in the element menu) that is combined with the Option and Command keys to create a direct shortcut. For example, the Parenthetical format is the fourth available element, and so it can be invoked with ⌥⌘−4; a Transition is the 6th element, and so can be invoked with ⌥⌘−6 and so on. Learning these numbers can be useful as you do not need to call up the intermediary menu to change element modes.

Upon hitting enter and typing, the paragraph formatting will be changed to that of the next set element. So, for instance, if you hit Enter after having typed in a Scene Heading, the typing attributes might be set to Action. At this point, you could easily change the current element by pressing the Tab key—hitting it once would change the current element to Character. Hitting it several times will cycle through different elements.

The footer bar tells you which element will be created next by hitting Tab or Return. For example, in the Screenplay format, when waiting to type in a character name, pressing Tab will switch you to Transition and pressing Enter will switch to Dialogue. However, once you begin typing in a character name, the tab hint will change to Parenthetical, letting you drop to that element instead of dialogue if you wish.

If for some reason you need to change the element type for text you've already entered, you can place the caret anywhere within the text you want to change, and invoke the element menu in the footer bar, or use the Format ▶ Scriptwriting ▶ Change Element To ▶ sub-menu to select a different element. Note that when moving from an element that uses all-caps to an element that does not, you will need to adjust the capitalisation manually, as Scrivener will not try to guess what is appropriate. The Format ▶ Convert ▶ sub-menu has a number of conversion routines to aid in this.

20.1.1 Auto-Completion

Using the built-in script settings, auto-completion is available for some elements, as is appropriate for their context. Upon choosing Scene Heading, for instance, you can start typing with "E" and you will be presented with a options like "EXT." and "EVENING". You may disable or enable (or add your own) additional items from this list by editing your script settings (section 20.7.1), though note you should in general not use this list to assign project specific completions like scene locations. It is best to use the Project ▶ Auto-Complete List... window for this. You may want to save your scripting adjustments for future use in another project. The auto-completion lists in the script settings should be seen as integral parts of the scriptwriting process in general.

Some elements have been configured to automatically do this for you. Scene headings are a good example. They will add anything you type in between a '.' and a '-' in a scene heading line. In practice this means the part marked in bold in this example would be added to the project auto-complete list: "EXT. **THE RED LION** - NIGHT". When

words and phrases are added to the project auto-complete list in this fashion, they will be assigned a scope (subsection 15.7.3) which restricts their eligibility to only when you are typing within that element. A character name will not suddenly pop up in the slugline, for instance. You can adjust the scope of automatically added phrases using the main project auto-complete list.

20.1.2 Dual Dialogue

If you intend to use the Final Draft FDX compile format for exporting, you can designate text as being dual dialogue by surrounding both bits of dialogue (including the character names) in a **Format ► Formatting ► Preserve Formatting** block. This will not negatively impact the text if you decided to use PDF or another format for proofing, prior to moving to Final Draft. Under normal conditions, it merely informs the compiler that the text in the blue box should not have its formatting altered. Since scripts are already pre-formatted, the compiler will not typically be touching the format anyway, and thus the designation does nothing.

20.2 Using Page View to Estimate Page Counts

While the point should remained stressed that Scrivener is not designed at any level to provide a completely accurate pagination solution, with rigid formatting, such as that used by most scriptwriting formats, and a few optional settings, it is possible to get very close to an accurate page count, when using Page View (chapter 16) mode.

The first thing you will need to do is switch Scrivener's default Scrivenings separator from a divider line to single line breaks, which can be set in the Formatting preference tab (subsection B.8.3), at the very bottom of the panel. This version uses a "zero-height" display model which will not introduce page inflation over long documents, like the standard divider will. You will also want to disable the usage of titles in Scrivenings mode, using **Format ► Options ► Show Titles in Scrivenings**, for your project, as it will likewise expand the overall height of the document over large sections.

> **Scrivenings View and Scriptwriting**: When creating a scrivenings session, if the session includes a mix of script formatting documents and regular documents, Scrivener will determine which mode to use based on the type which has the most entries in the session. This means, if you create a scrivenings session with 4 standard documents and 2 script documents, script mode will not be on by default. In the other direction, if there are more scripts that standard documents, script writing tools will be enabled in those standard documents. You can switch modes while in scrivenings with the **⌘–8** shortcut. This will only impact (globally) the scrivenings session, and not any of the underlying files.

Finally, you will need to ensure that the paper size is set up correctly in the File ► Page Setup... dialogue.

If you work with scripting formats quite a lot, you might wish to ensure that "Show page view in new projects" is enabled in the Editor preference tab as well.

20.3 Importing a Script from Final Draft and Other Programs

> **Final Draft Users**: Refer to the section (section 20.5) on exchanging files with Final Draft, version 8 and greater. For users of Final Draft 7 and earlier, you can import scripts created in Final Draft by using "Save As" in Final Draft to save your script in "File Converter" (FCF) format, then import the resulting .fcf file. Scrivener will try to match the elements in the FCF script to the elements in the current script format. For projects that use the basic screenplay formats, this should be all you need to do to import a script with all of its elements recognised.

Because Scrivener is not a dedicated scriptwriting program, the way it handles script elements is different from Final Draft or Movie Magic Screenwriter. Dedicated scriptwriting programs generally assign hidden meta-data to each text element that tell the program that, for instance, this line of text is a scene heading, that paragraph is a piece of dialogue and so on. Scrivener does little more than automate the process you would have to use yourself were you to use a standard word processor to format your script; that is, it just sets up the necessary paragraph formatting for each element and auto-capitalises if necessary. Scrivener recognises elements by their paragraph formatting. For instance, if a paragraph has a three-inch left indent and single line spacing, Scrivener will look up this formatting in the list of script elements, and if an element is found with matching formatting its name will be selected in the elements pop-up button in the footer view. If no elements match, "General Text" will be selected.

This means that if you import a script into Scrivener from another program and want to continue working on it, the script format mode selected in Scrivener must exactly match the formatting of the script you have imported.

20.3.1 Importing From Final Draft 7 and Earlier

First, you will need to export the script from Final Draft as an RTF file. Follow the instructions provided with your software package, for doing so.

Import the script either by dragging the RTF file from the Finder into Scrivener's binder or by using File ► Import ► Files... Select "Screenplay" from the Format ► Scriptwriting menu.

Ensuring that the script is visible in the editor and that the editor has the keyboard focus (by clicking into it if necessary), go to **Format ► Scriptwriting ► Re-capitalize Script** to capitalise scene heading and character elements.

The script should now look right in Scrivener, and clicking into different parts of the text should cause the pop-up button in the editor's footer view to update to display the element in which the cursor has been placed (e.g. "Character" if you have clicked in a character element).

If the pop-up button in the right of the footer view displays only "General Text" when you click into different areas of your script, or if "Re-capitalize Script" did not work properly, then this is because the format mode you selected ("Screenplay" in this example) does not exactly match the format of the script you imported.

20.3.2 Importing Other Scripts as RTF

If your script formatting was not recognised properly, or if you imported a script that has no matching format mode in Scrivener, you will need to create your own format mode that matches the script using the **Format ► Scriptwriting ► Script Settings** panel.

This method can also be used to recover Scrivener formatted script files, if the original scripting settings have been lost for one reason or another.

Here's how:

1. Export the script from your scriptwriting program as an RTF file.
2. Import the script either by dragging the RTF file from the Finder into Scrivener's binder or by using **File ► Import ► Files...**
3. Ensuring that the script is visible in the editor, click into a line of text that represents one of the elements you want to be recognised (for instance, by clicking into a line of text that should be a scene heading).
4. Open the Script Settings panel (**Format ► Scriptwriting ► Script Settings...**).
5. Enter a title for your new format in the "Format Title" text field.
6. Select the first element in the list and give it the name you require (i.e. the name of the element in which you placed the cursor in step 3) by double-clicking into it and editing it if necessary.
7. From the **Manage...** drop-down menu in the Script Settings panel, select "Use current font & paragraph settings". This will copy the font and paragraph information from the line of text in which the cursor has been placed in the editor into the Font and Paragraph panes of the Script Settings sheet.

Repeat this process, matching the settings for the elements in the Script Settings panel with the text in the editor for each element in your script.

Be sure to save your script format for use with other projects using the **Manage...** pop-up button at the bottom of the Script Settings panel.

You can then use your new format with all projects in the future for any scripts you import. Once you have successfully created your own script format mode, all of the elements in your imported script should be recognised in the pop-up button in the footer view of the editor, and you should be able to use the script mode to continue editing your script. Note that more complicated script formats may require a little more tweaking to be recognised, but the above process should work for most.

If you do create a script format mode that recognises elements from an imported script, please use "Save to file..." in the **Manage...** pop-up menu from the Script Settings panel to save your format as an XML file somewhere safe on your hard drive so that you can back it up and load it on other machines or following a hard drive reformat if necessary. Also, please feel free to upload such format files on the forums, where other Scrivener users may find them useful—or from where you can always download them again should you lose them.

20.4 Printing or Exporting a Script

You can use the compiler (**File ► Compile...**) to print your script (or to turn it into a PDF file). For examples of how to do this, take a look at one of the scriptwriting project templates by going to **File ► New Project...** and selecting a project template such as the Screenplay template. Read the instructions that come with the template to see how to set up your project so that it is formatted properly when printed (or exported).

In the majority of cases, just as you would with many other types of writing in Scrivener, you will want to export your script from Scrivener so that you can do all the final formatting in a dedicated formatting program such as Final Draft, see Exporting Scripts (subsection 24.3.4) for information on how to use the Compiler to export to many popular scriptwriting programs.

20.5 Working with Final Draft

That this information is only relevant to Final Draft 8 users. Users of previous versions of Final Draft should use the FCF format for importing and exporting their work.

20.5.1 Importing Formatting from a Final Draft Document

> **Optional Step**: For basic scripts or those that will use Final Draft's standard screenplay formatting—indeed, for most screenplays—this step can be omitted, as it is mainly concerned with setting up the formatting for custom scripts. Proceed to the next section if this is the case.

Whenever you import a script from Final Draft into Scrivener, all of its elements will use the formatting specified in Scrivener's script settings. These can be set up by selecting "Script Settings…" from the Scriptwriting submenu of the Text menu. The Script Settings panel should be familiar to anyone who uses Final Draft, as it is set up to be very similar, allowing users to create custom script formats.

Because a Scrivener project comprises multiple documents, it is important that the script settings are set up as the user desires *before* importing or creating any script documents in the project.

If you are writing a screenplay using standard formatting, you don't really need to do anything here. However, if you are using a custom format, or if you have an FDX script that uses a custom format which you wish to retain after import into Scrivener, you need to set up Scrivener to use that custom format first.

Fortunately, this is very easy to do, as Scrivener can read the formatting directly from an FDX file, but you must do this before importing the script itself. (Remember, Scrivener can hold many different script documents but can only use one script format in a project at a time; this is why setting up the format must be done separately from importing a script.)

To import the formatting from an FDX file:

1. Open the Script Settings panel by going to **Format ► Scriptwriting ► Script Settings...**
2. In the Script Settings panel, click on the "Manage…" pop-up button in the bottom-left.
3. Select "Load from Final Draft .fdx or .fdxt…"
4. In the open panel that appears, select the FDX file that contains the elements and formatting you want to use and then click "OK" and "OK" again on the Script Settings panel to accept the changes.

This doesn't import the actual script, it just imports its elements and formatting for use in the current project.

To test the new format, create a new document, choose scriptwriting format (**Format ► Scriptwriting ► Script Mode - FDX Script Settings**) and start typing (changing elements using tab and enter or using the pop-up menu in the footer view beneath the editor). You will see that the script uses the formatting of the FDX file. Now that this is set up, you are ready to import the contents of the FDX file.

To save the format for future use in other projects, use the same "Manage…" drop-down menu to select "Save for use with other projects". The imported script format will now appear in the main **Format ► Scriptwriting ►** sub-menu for all projects.

20.5.2 Importing FDX Files

You can import FDX files into Scrivener in one of two ways. Both are fully featured, the only difference between the two is that the second will offer options for cutting up the script into smaller pieces automatically:

1. Using the standard file import methods; drag and drop into the project binder, and **File ► Import ► Files...**.
2. Go to **File ► Import ► Import and Split...** and select the FDX file that you want to import. See Import and Split (subsection 11.1.7) for further details.

Many features will be retained: script notes become Scrivener footnotes, revisions are marked up in red, highlighting is retained and so on.

If you wish the FDX file to be part of the script that will eventually be exported, be sure to import it into the "Draft" folder, or drag it there after importing.

20.5.3 Exporting Individual Documents to FDX

To export individual scripting documents to Final Draft, just select the documents you want to export in the binder and then go to **File ► Export ► Files...** Select "Final Draft (FDX)" as the export format.

The exported file should open in Final Draft with all features intact.

More commonly you will want to combine all of the scripting documents in the Draft folder of your project into one FDX file:

1. Go to **File ► Compile...**
2. Select "Final Draft (FDX)" as the file format.
3. You may need to tweak the various settings of the Compile Draft sheet to get the script formatted correctly, just as you would with any Scrivener project.

4. Click on "Compile…" and choose a filename.

5. Open the exported file in Final Draft.

You should see that most elements are retained—footnotes become script notes, coloured text becomes revised text, and so on. The header shows whatever is set in Scrivener's Compile Draft panel.

That covers the basics of importing from and exporting to the FDX format using Scrivener.

20.6 Working with Fountain

Fountain[2] is a simple markup language (based loosely upon Markdown[3]) designed just for screenwriters who need or prefer to work in a plain-text environment (which will be of particular interest to scriptwriters who prefer to write on the go with a mobile device). Fountain helps with composing and editing without losing the individual meanings behind the elements (character name, action, transition, etc.). From the Fountain website:

> Fountain is a simple markup syntax for writing, editing and sharing screenplays in plain, human-readable text. Fountain allows you to work on your screenplay anywhere, on any computer or tablet, using any software that edits text files.
>
> Taking its cues from John Gruber's Markdown, Fountain files are eminently readable. When special syntax is required, it is straightforward and intuitive.
>
> Even when viewed as plain text, your screenplay feels like a screenplay.
>
> Fountain supports everything a screenwriter is likely to need in the early, creative phases of writing.

When importing or syncing Fountain files, they will be processed and converted to Scrivener's internal scriptwriting system for you. There is no need to use the markup conventions while writing within Scrivener. Certain Fountain conventions will be converted to rich text or Scrivener meta-data, in addition to the scriptwriting elements themselves:

- *Boneyard markers*: sections of text marked between "/* boneyard markers */" will be imported into Scrivener as struck-through text and vice versa.

[2] http://www.fountain.io/

[3] http://daringfireball.net/projects/markdown/

- *Dual dialogue*: when using the syntax for dual dialogue, Scrivener's convention of placing the characters and dialogue in question in a Preserve Formatting block. Likewise adjacent dialogue marked with Preserve Formatting will be marked in exported Fountain files.

- *Script notes*: when importing, text marked as a script note that appears entirely on its own line(s) will be placed as an inline annotation. Otherwise, when script notes are added to text that is otherwise an element, they will be anchored to the nearest word and linked as comments in the Inspector. If you prefer one form of annotation over another, you can use the Format ► Convert ► sub-menu to switch between styles. Scrivener uses inline annotations in cases where script notes are on their own paragraph to avoid potential problems with removing the text and causing elements around that text to lose their syntax. For export, all annotations and inspector comments will be converted to script notes, if you opted to include them in your settings.

Fountain is for Screenwriters: One important detail to remember is that Fountain has been designed for screenplays. Using it with other scriptwriting formats is not supported as it may work unreliably or not at all. If you do wish to try using Fountain with another scriptwriting format, all you need to do is make sure that the elements (section 20.7) are named precisely as they would be in the standard screenplay format.

Scrivener can both import the Fountain format two different ways:

- *As a single script*: merely drop the Fountain file into the binder. Scrivener looks for the '.fountain' extension, so make sure the file has had its extension changed accordingly, otherwise the file will be handled as plain-text. When properly handled, all elements should be imported and formatted automatically to the screenplay script format.

- *Split up into scenes*: as with Final Draft files, you can use File ► Import ► Import & Split... to select the file. Again, it will need a '.fountain' extension to be properly recognised and imported. This will result in a new binder item being created for each discovered slug-line. If a section header has been typed in immediately prior to the slug-line, it will be used to populate the binder item title[4]. Fountain synopses texts will be gathered from throughout the scene and combined into Scrivener's own synopsis index card. Read more about Import and Split (subsection 11.1.7).

[4]Note that while Fountain has support for multiple levels of depth with its header syntax, Scrivener's scene import will produce a flat list.

For integrating with a mobile device, Fountain is a selectable format in the folder synchronisation feature (section 13.2).

Finally, it is possible to compile to the Fountain format (subsection 24.3.4).

20.7 Creating Your Own Script Formats

Scrivener comes with a number of script formats built in. You can also create your own script formats tailored to your own requirements; as well as export and import script modes for sharing on the Internet.

To create your own script format mode, select **Format ▶ Scriptwriting ▶ Script Settings...** and use the window that appears to name your format and define its various elements. At the top of the panel is the Format Title text field. Enter the name of your format here (e.g. "Television (BBC Comedy)"). This is the name which will be visible in the menu, if you later choose to save your settings to a script format that other projects can use as well.

On the left of the panel is the Elements list. This is a list of the parts that make up your script, such as "Scene Heading", "Dialogue" and so on. You can add new elements and delete existing ones using the **+** and **-** buttons beneath the list, and you can rename elements by double-clicking them. Reorder elements by dragging and dropping them; this will change their ⌥⌘–# assignment (where '#' corresponds to the numerical order in which the element appears). Note that "General Text" and "General Text (Centered)" will be added to the end of any element list you create but will not appear here as they have no special properties by definition.

Next to the "Manage..." menu (subsection 20.7.2) is another drop-down menu that allows you to choose the units used to define your style: inches, centimetres or points (most script formats use inches, of course).

On the bottom-right of the panel are the "OK" and "Cancel" buttons for saving or cancelling the changes you have made to the format mode respectively. Clicking **OK** saves the script format into the *current project package* only. This means that if you share your .scriv file with someone else, that someone else will be able to open the project and use the script format regardless of whether they have this script format installed on their machine. The format will not be available for use with other projects until you select "Save for use with other projects" from the "Manage..." menu. Once you are happy with the format, it is thus recommended that you make it available for use with other projects, as you might not remember which project file you saved it in.

It's worth reiterating that script settings are saved directly into the project itself automatically, when you click the **OK** button. There is no need to repeatedly save a customised

script. Saving a script is for when you want to use that format for another project entirely, or to a file for transport to other computers.

> **Pro Tip**: Making a format available for use in other projects saves it as an XML file in the ScriptFormats folder within the Scrivener support folder (use File ► Reveal Support Folder in Finder if you are unsure of where this is). Consider instructing your backup software to include this location (and potentially others as well within the Scrivener folder) in order to keep your settings safe. You can also save out external copies for sharing or further safe-keeping, with the Manage… menu, discussed below.

20.7.1 Format Tabs

On the right of the panel is the tab view. This is used for setting the format for each element. Select the element from the list on the left, and then set the formatting for that element using the controls in the tab view on the right. The setting controls are grouped under four different tabs, which this section will describe.

Font Tab

The Font tab provides options for setting the character appearance for the selected element, as follows:

Font Allows you to set the font for the current element (selected in the list on the left). Scrivener does not use the font to identify elements, so you can change the font in the preferences without messing up Scrivener's recognition of script elements. This also holds true for font changes made in the editor.

Size Allows you to set the font size in points.

Style Provides a number of options for determining the appearance of the current element, mostly self-explanatory.

- *All caps*: If this is checked, the element will be capitalised. If "End of line" is selected, the element will be capitalised all the way to the end of the element; for the purposes of this definition, a paragraph is a line. If "First tab" is selected, the element will be capitalised only up to the first tab character. This accommodates formats such as the UK stage play format, which has character names and dialogue on the same line, with character names in capital letters followed by a tab and dialogue in normal sentence case. If "Character(s)" is selected, you can enter the keyboard characters into the text field that will terminate capitalisation. For instance, if you entered a colon in the text field, the element would only be capitalised up to the first colon.

- *No Underline/Single Underline/Double Underline*: allows you to set the underline style for the current element. If single or double underline is selected, you can check Underline by word to underline words only (not spaces).

Colour and Adornment Set the text colour used for this element in the text editor, or automatically add parentheses around the text you enter.

- *Parenthetical*: encloses the current element in brackets when you select that type.
- *Color*: The text falling within the range of this element's definition will be recoloured as specified here. To change the colour, click on the chip and select a colour using the palette.

Menu shortcut Enter the single letter here that will be used in the script elements menu as a shortcut key.

Paragraph Tab

The paragraph tab provides paragraph formatting for the selected element:

Alignment Sets the current paragraph alignment (left, centred, right or justified).

Spacing Sets the inter-line spacing (single, 1.5 or double).

Spacing Before Sets the number of blank lines to appear between the selected element (0, 1, 2, or 3), and the last one above it.

> **Ruler Conversions**: When working with units in Scrivener, keep in mind that its ruler starts at margin zero instead of paper zero. Since Scrivener is, by and large, not "aware" of paper settings and page layout, it counts its ruler settings from the beginning of the text on the left end of the page, not the beginning of the paper itself. This is in contrast to many word processors, which start measuring at the paper left edge, and show the print margin buffer in the display of the page.
>
> Consequently, to convert most standard measurements to useful values here, you will need to factor in the standard amount of print margin. For example, if the Scene Heading is specified to begin at 1.5", you will need to subtract 1" from that and input 0.5" into Scrivener, since an additional inch will be added to the layout, once margins are added to the page outside of Scrivener.

Indentation Allows simple customisation of the ruler layout for each element, providing support for most if not all formatting requirements on the page.

- *First Line Indent* Sets the left indentation of the first line of the paragraph, a value which will only impact the first line and no subsequent lines from the same paragraph.
- *Left Indent*: Sets the left indentation of all lines in the paragraph excepting the first. Set this value identical to the First Line Indent, to create a uniform "block indent" look.
- *Right Indent*: Sets the right indentation of the paragraph. This setting pertains to all lines.
- *First Tab*: Sets the first tab stop of the paragraph. (Note that subsequent tab stops may be added automatically to differentiate elements should any elements have the same formatting.)

Advanced: Allows you to set the writing direction, minimum line height and default tab interval (the latter for cases where extra tabs are added as mentioned in "First Tab", above). Set "Try to keep with next paragraph" to prevent widows and orphans with this element.

Tab/Return Tab

The Tab/Return pane allows you to control the tab and return behaviour for the selected element; that is, it lets you specify what happens when you hit the tab or return keys, and so thus can be used to aid in the flow of writing. If you are creating a scripting environment from scratch, you might wish to save this step for last, since you will need to reference other elements (which may not exist while you are going through the list, initially).

On return Sets which element formatting the text will use when you hit the Return key. Using Screenplay as an example, for the "Scene Heading" element, "Action" is selected for "On return". This means that if you hit the return key after typing a Scene Heading, the text will automatically be formatted and ready to type an Action element.

Tab behavior Sets what happens when you hit the tab key.

- *Allow tabs*: If this is checked, the tab key works as it would normally, that is, it inserts tabs. With this checked, none of the other options for tab behaviour are available.
- *Tabbing on an empty line*: Choose an element from the "Go to:" pop-up button to set which element will be formatted when you hit tab on an empty

line. This allows you to cycle through elements using the tab key. For instance, in the Screenplay format, if you are at the beginning of the line using the Scene Heading format, you can then hit the tab key to reformat the line as an Action element provided you haven't typed anything yet.

- *Tabbing after typing*: Sets what happens when you hit the tab key after typing something in the current element. Select "Go to:" and an element from the pop-up button to have tab automatically take you to the next desired element (inserting a new line automatically). Use this to define convenient alternate common elements. Select "Insert" and type characters in the text field to have the tab key insert a character sequence. In the example above, hitting tab after typing something in a Scene Heading element automatically inserts a hyphen surrounded by spaces, so you can easily enter a chronological marker.

Auto-Complete Tab

The Auto-Complete tab allows you to set a custom list of words that will appear for auto-completion while typing in the current element.

Use the + and - buttons below the list to add or remove new words to the auto-complete list, and double-click on words to edit them. When typing your script, hit ⌥Esc or ⌘–. to bring up the suggestion list beneath your cursor. Additionally, if you have your options set up to offer suggestion automatically while you type, and the first letters typed in will match a phrase in this list, you will be presented with the auto-complete list. Note that the auto-complete list only appears after you start typing a word. For each entry, you can check the "Go to Next Line" box, which will force the editor to move to the next element (as it would if you pressed Return) once it has been entered.

Append project auto-complete list Adds any words set in the project auto-complete list (Project ▸ Auto-Complete List) to the current script auto-complete list. This way, script element completion lists can include character names set at the project level.

Automatically add phrases to project list that occur Type in two characters to help in isolating useful part of a phrase (such as locations and character names) from the paraphernalia surrounding them. If a field is left blank, then anything on the line will be used, up to any character defined in the opposing field.

An example for the Character element would be to add anything typed into the character field prior to an open parentheses, and thus avoids such common markers as "(O.C.)". However no starting limiter is provided, so anything typed into the line will be added to the project list.

After project list completions When a project list auto-completion has been used, you can set up the scripting system to do nothing, go to the next line, or insert a tab.

An example for the Character element would be to go to next line after using a project list completion (most likely a character name).

20.7.2 Managing Scripts

At the bottom-left of the main script settings panel is a drop-down menu entitled "Manage…". This allows you to reset your script elements to the default screenplay settings, to save your format for use with other projects, to save it for back-up somewhere on your hard drive (or to share it with others) and to load from file an existing format mode created either by yourself or someone else. You may also import formats from Final Draft FDX or FDXT files.

Reset to defaults Reverts all customisations that you have made to the settings panel to the Screenplay default. If you wish to revert to a saved script setting, simply select that script from the Scriptwriting menu, instead of using this tool.

Use current font & paragraph settings Will attempt to import all available character and paragraph level formatting attributes into the *currently selected* element in the above element table. This can be quite useful when you have imported a script from another program and wish to create a script format from the existing text. Simple click through the document, locating element types, and use this tool to import the correct formatting into each element type. If you are using Final Draft 8 or above, you will want to use the "Load from Final Draft" tool, mentioned below, for scanning .fdx files for types, rather than doing all of this by hand.

Load from file… Loads a Scrivener script format file from the disk. Use this if you have downloaded a script format from the Internet, or are in the process of transferring formats from one computer to another.

Load from Final Draft .fdx or .fdxt file… Scrivener can examine an existing Final Draft or Final Draft Template file for formatting rules and names, and attempt to convert them to Scrivener's internal script formatting, in one convenient shot.

Save to file… Saves your current settings to an external file that you can easily backup, upload to the Internet for sharing, or send to another of your computers.MAC: This will not install the saved script into your Scrivener support folder. Use the below command for that.

Save for use with other projects Essentially accomplishes the above, but automatically installs it into your Scrivener support folder for immediate usage in all of your projects. If you wish to share the file with others, or transfer them to another computer, use the Save to file... option as well as this one.

20.8 Using Script Formatting for Other Purposes

While the Scriptwriting mode has been preconfigured and designed for scriptwriting, it deserves to be mentioned that it is at its heart nothing more than an automated styling engine. This means it can be used for a variety of purposes having nothing to do with scriptwriting.

For instance, you could use it to aid in the transcription of an interview. By setting up formatting "elements" for each participant in the conversation, you could easily attribute each portion of text to the appropriate speaker, and if the interview is a back and forth between two individuals, you could even set it up to alternate back and forth between speakers whenever you press return.

If you need to create a structural document, and cascading between structural types would be of benefit, then consider using the Scriptwriting engine to create you own format from scratch.

Chapter 21

Writing Tools

21.1 Searching and Replacing

There are three primary tools for searching for items in a project. Project searching has numerous options for searching for particular types of information (like say, searching just in synopses), and will always scan the entire project for matches. Document searching is more familiar in that it provides a Search and Replace window that will act on the current text you have loaded. Third, format searching has many ways to search for particular types of formatting in your project.

Depending on the type of search being used, you will be faced with two different types of response from the application. Project searching *gathers* all documents that match the criteria together, and displays them in the sidebar. These search results can be used as the basis for collections, either as a selection for a new standard collection, or to save the search criteria as a saved search collection. Holding search results in the sidebar also means you get to use all of the document management tools you are familiar with by now.

Document and format searching on the other hand do not collect results, but rather step through the search results one by one, in a familiar fashion. Document searches will never "escape" the text editor. That is, they will not go off into other documents you do not have loaded, looking for results. Format searching, as with document searching, operates in a step-by-step manner, but will by default step through every matching item in the entire project.

21.1.1 Document Find and Replace

As with many Mac OS X applications, you are provided with a standard Find and Replace panel which can be called up in any text view with ⌘–F. The find panel operates only in the current text editor (which might include multiple documents when using Scrivenings mode). To replace text in all documents use Project Replace (see below); to search for text in all documents, use Project Search.

Find (and replace) will function on any comments or footnotes in the Inspector pane (section 18.2). Replace will always work in the background, but for Find to highlight

matches, the Inspector must already be open.

You can use regular expressions for both the initial search term and its replacement text, by enabling RegEx mode in the "Find Options" drop-down menu. When using replacements, if you wish to refer to a stored value from the initial search pattern, use the backslash-numeral syntax for doing so.

In addition to the buttons available in the panel, there are some keyboard shortcuts you can learn which can reduce the reliance upon clicking within the panel to carry out searches.

- **Return**: is nearly synonymous with clicking the "Next" button, with one important difference, the Find window will be automatically dismissed after you press Return. The search term will be saved however, allowing you to continue using the two following shortcuts without any interface in your way
- ⌘–**G**: finds the next available match; will wrap around to the top of the document if there are no more matches available below the current point. This command works even if the Find panel is closed, and it will operate off of the last active editor split (even if that split is not focussed, though searching will move your keyboard focus to the editor)
- ⇧⌘–**G**: finds the previous match; as above, but will wrap around to the bottom of the document. As with forward searching, this command uses the last active split and will focus the editor as necessary.

Additionally, ⌘–**E** can be used to load the currently selected text into the "Find" field without opening the panel. Using this method, you can define a search term and then use ⌘–**G** to find the next instance of that term without even opening the panel at all.

While the Find panel is open, you can continue editing in the main project window. You may also restrict matches to only those which have the same letter case, are look for matches which only fall within the start of a phrase, end of a phrase, be a whole word, or the default, "contains" which matches even parts of words.

The **Next** and **Previous** buttons will jump from one result to the next. When the last match is discovered, the Find tool will wrap around to the beginning of the document again.

There are three replacement options.

- Replace: replaces the currently matched item
- Replace & Find: equivalent to clicking the **Replace** button and then the **Next** button

- Replace All: will replace all matches with the provided replacement value with no further interaction.

To leave the Find tool, you can either click the close window button, or tap the **Esc** key.

21.1.2 Project Search

Project search is a tool for finding binder items which match the search you've provided. This is unlike the standard Find tool, which steps through a document one match after another. Instead, it returns everything at once in a list, which you can then go through at your leisure. You can think of it as a way of "filtering" the Binder so that it only shows those items which match the search.

As a convenience, the search text you provided will be automatically loaded into the Find search tool for you. This means you can immediately use the keyboard shortcut to find the next search result within the document, ⌘–**G**. Note however that because Project Search can operate on many different types of item elements, the search result itself may not be in the primary editor. It might be a label, or a keyword, instead. In these cases, ⌘–**G** will not located the search result for you.

When you use project search, the results will be placed into a temporary, dedicated collection called "Search Results". If the collections interface has been hidden, this fact will be largely obscured from you (though the background colour of the sidebar will change, as will the title for the sidebar, at the very top), but the collection will exist, and can be accessed later (even after closing the project) so long as you do not run another search.

> **Search Results and the Sidebar**: Since search results temporarily replace the binder in the sidebar, it can initially confusing to figure out how to get back to the binder. When the collection interface is revealed, you can switch back easily by clicking on the "Binder" tab, but otherwise you can use the small **X** button in the sidebar footer, or simply tap **Esc** while in the search field to dismiss the search results. For more about navigating amongst collections in general, read Using Collections (section 8.4).

By default any search result list (including saved search collections) will show label and status for you in columns as a convenience. If you would rather a cleaner display of just the names, right-click anywhere in the column header bar and select "Hide Label & Status Columns".

There are two ways to access Project Search. You can either click in the search field in the toolbar, or press ^⌥**F** to place the cursor focus there. If the toolbar has been hidden,

or set to text only, a convenience window will be opened providing the same features as the toolbar search.

When nothing is entered in the search field, greyed out text will inform you of the current search mode. By default this is "All (Exact Phrase)", which means all searchable elements in the project will be analysed with the "exact phrase" operator, which means that if you type in two words, they will need to appear together in that order. The search scope, data type, and operator mode can be adjusted via the project search options menu, by clicking on the magnifying glass to the left of the search field.

The following sections will describe each of the available options to you within this "magnifying glass" menu.

> **Finding (Almost) Everything**: While it may at first seem a bit useless to use search to find everything in the Binder, it can often be useful as searching this way presents your entire project in a flat list. In conjunction with sorting by columns in the Outliner, this can reveal useful information, such as sorting by Modified Date to see everything you've recently edited. To search for everything, simply type in a single asterisk (*) into the search bar.
>
> Another use for this feature can be to combine it with some of the other search options to return everything that otherwise matches a specific criteria. Hence, "almost" everything. If you leave the scope and field options to the most permissive, then searching for an asterisk will find everything in the project—even things in the Trash. However, if you change the search scope to only operate in the Draft folder, for example, an asterisk will only return all Draft items. Searching the "Label" field for an asterisk will only return items which have had a label assigned to them.

Search In

Sets the data type of the search. You can set the search to only query a certain type of meta-data or text field, rather than everything. You can also search within multiple field types at once by holding down the Option key and selecting another type (though of course if "All" is selected this will only result in changing the type to the selected search field, you'll need to use the Option key on a second field to add another). Option selecting can also be used to turn off an extra field without disturbing the total selection. If you wish to fully reset the search criteria to a single source, either select All, or select any field without using the Option key.

- *All*: the default search mode. Every available type of searchable field will be queried for matches

- *Title*: only the titles of binder items will be searched for. This is similar to title filtering in the binder
- *Text*: the text contents of files and folders will be queried. Note this does not include notes and synopsis
- *Notes*: the auxiliary document notes will be searched. Note this does not include project notes
- *Synopsis*: the synopsis field for each document will be searched. This is anything that has been typed into the text area of an index card or in the synopsis portion of the outliner
- *Keywords*: only keywords will be searched. Note you can also perform keyword searches quickly by using the Project Keywords window
- *Label*: the text (not colour) of the label meta-data will be searched. Note the name of this meta-data field can be changed per project
- *Status*: the text of the status meta-data will be searched. Note the name of this meta-data field can be changed per project
- *Custom-Meta-Data*: the text values (not meta-data field name) will be searched.

Operator

Set the method by which search terms will be handled as you type them in.

- *Exact Phrase*: the default method. What is typed into the field will be queried precisely as it is typed in. "the book" will only match documents that have the phrase "the book" as written, not documents that just have the word "book". It will also return documents that contain "the books". For exclusive matching, use Whole Word (below).
- *All Words*: every word entered into the search field must appear in the selected data type. Documents which only match some of the words will not be returned. Words can be entered in any order. Analogous to logical AND.
- *Any Word*: queried documents must contain at least one of the words typed into the search field. Analogous to logical OR.
- *Whole Word*: unlike any of the above search methods, the term supplied will only match whole words. A search for "Jo" will only return documents with that word, not documents that also contain "Joseph".

- *RegEx*: enables the advanced regular expression syntax (subsection 21.1.5).

Options

Provides a few extra options, as well as searching scope limiters. Scope limiters instruct the search engine to only scan parts of your binder, rather than the entire thing.

- *Search Draft Only*: will only look in the Draft folder of the binder. Note that if the name of Draft has been changed in the project, the title of this option will reflect that name change
- *Search Binder Selection Only*: pre-select items in the binder and then perform the search against those items only. This selection is explicit, not implicit. Selecting a folder will not include all of its children in the search query as well. This option is not used in Saved Search Collections
- *Exclude Trash Documents*: ordinarily, items located in the trash will be discovered by search results, using faded icons in the search results list. If you'd rather not see them at all, use this option to disable Trash folder searching. This option is not used in Saved Search Collections
- *Search "Included" Documents*: include documents that have been marked as "Include in Compile" in their meta-data. This option can be combined with the below
- *Search "Excluded" Documents*: include documents that have *not* been marked as "Include in Compile" in their meta-data. This option can be combined with the above
- *Case Sensitive*: by default, the search engine ignores letter case. If you need to search for proper nouns and are getting a lot of false positives, this option can help

Save Search As Collection...

Creates a special Saved Search Result Collections (subsection 8.4.4), which will keep itself dynamically updated whenever you view them.

21.1.3 Project Replace

The menu command, **Edit ► Find ► Project Replace...** provides the ability to replace text throughout the entire project. Set the text to find, the text with which to replace it, whether the search should be case-sensitive or apply to whole words only and specify what should be affected by the replace (any combination of text, titles, notes, synopses and snapshots). Note that a Project Replace cannot be undone (except by using "swap"

to exchange the search term for the replacement text and repeating the Replace All operation, which in some cases might have unexpected results). A progress bar at the bottom of the sheet shows you the progress of the replacement operation—bear in mind that it could take a little while on large projects.

As with all tools that make widespread changes in a fashion which cannot be undone, it is often a good idea to back up your project prior to use it. A simple mistake can render your entire draft illegible, or even worse, produce subtle flaws that even your editor ends up missing. Using the "Whole words only" option can mitigate this a great deal. Without that option, searching for the character name "Sam" and replacing it with "Joseph" could end you up with such (bleakly) amusing concoctions as "Josephe" in place of every "same".

You may use Regular Expressions (subsection 21.1.5) for both the Replace and With fields, if you prefer. This may cause the operation to take significantly longer, depending upon the complexity of the expression.

You can choose the scope to optionally affect a variety of document meta-data, as well as just the text of a document. The following options are available:

1. Text: standard text editor contents (including any linked footnotes and comments from within that text)
2. Titles: the titles of the documents as they appear in the binder
3. Notes: any inspector note fields that are attached to documents
4. Synopses: the text content of each index card
5. Snapshots: archived snapshots will be changed, as well (take special care with this one, as snapshots are your internal backup mechanism) as any linked footnotes and comments originating from those snapshots
6. Project notes: any project notes will be impacted as well.

21.1.4 Find Synopsis

Sometimes you just want to jot down a quick idea or note to yourself on a particular item within your project, but you don't want to hunt around for it in the Binder (maybe it isn't even visible), or use Project Search and lose your current working environment. The synopsis, an integral component of every item in your project, is useful for this sort of task, and especially so when coupled with the Find Synopsis feature.

To call up the search tool, use the menu command, **Edit ► Find ► Find Synopsis**, or press **^⌘-G** from anywhere the main project window. An integrated search and edit window

will appear. As you type in the search field, the Title *and* Synopsis text for every item in the project will be scanned, and any matching items returned in the table below the search field. The search type used is phrase based. This means that the sequence of letters you type in must match either the title of the card, or any text within its synopsis. If you skip words or letters, the card will not be returned as a result. While searching, you can use the UpArrow and DownArrow keys to move the selection bar in the results table.

With the card selected that you wish to edit, press the Return key to jump immediately to the synopsis field, or use your mouse to click in either the title or synopsis area of the index card. Click on other entries in the tables to view other cards, or use ⌥⌘–**UpArrow** and ⌥⌘–**DownArrow** to flip between cards while you are editing in a text field. You may also use the Tab key to switch from the search field, to the results table, to the index card itself.

> **Pro Tip**: You can also call up the Find Synopsis window using ⌘–**F** when the application focus is in either the Corkboard or Outliner view.

If you wish to perform another search immediately, you can tap the Esc key to return to the search field at any time. Pressing it a second time will reset the search. Pressing Esc a third time, or in an empty search field, will close the window. The window can also be closed with the standard ⌘–**W** shortcut. Your last search will be remembered within the current session, so you can feel free to close the search window for a while, and recall it later without losing your place.

Sometimes a quick note isn't enough. If you want to transition from the index card to the item's QuickReference view, double-click on any of the search results, or tap the Spacebar while it is selected in the search result table. You may also use the standard **View ▸ Reveal in Binder** command to highlight any active card or selected search result, in your project binder. Items can be dragged from the result list into the header bar of any editor split to load it there. They can also be dragged into any Reference table in the Inspector, making this a handy way to look up and store references to items that relate to what you are currently editing.

When an item has been set to use a graphic to represent itself on corkboards, the image will appear in the card, rather than the synopsis text. This functions in the same way as the index card in the Inspector, in that you can drop graphics into it from either the Binder or your computer to update or add a graphic. To switch between graphic and text modes, right-click on the item in the search result list and select the appropriate option.

The Synopsis Finder can also tap into the regular expression syntax (subsection 21.1.5). Click on the magnifying glass icon menu beside the search field to switch this mode on.

21.1.5 Regular Expressions

[**10.7+ (Lion) Only**] Regular expressions are an advanced search syntax that can also be used to do complex replacements. If you would like to learn how to use them, there are plenty of recipes, learning resources and tutorials on the Web, as well as books that go into great detail. Scrivener uses the stock RegEx engine supplied by the Mac, which is based upon the Perl Compatible Regular Expressions (PCRE) guidelines, extended to provide UTF-8 support. Anyone who is familiar with using regular expressions in Perl will feel at home searching in Scrivener.

Regular expression support is provided in the following locations:

- *Project Search*: select "RegEx" from the operator section of the magnifying glass icon menu.
- *Project Replace*: using **Edit ► Find ► Project Replace...**, enable regular expressions by checking off "RegEx" in the list of provided search options.
- *Document Find*: the **Edit ► Find ► Find...** (⌘–F) panel supports them. Select "Regular Expressions (RegEx)" from the "Find Options" drop-down menu.
- *Synopsis Finder*: use the magnifying glass icon menu to switch to regular expression mode.
- *Compile Replacements*: RegEx has been added as a type of compile-time replacement. This will run when the project is fully exported, and will not alter the underlying text.

When used for replacing text in either the compile pane, or the document find panel, you will need to address stored variables using the dollarsign-numeral syntax, such as **$1** for the first stored variable, **$2** for the second and so forth.

21.1.6 Find by Format and Text Function

The Formatting Finder palette, **Edit ► Find ► Find by Formatting** (^⌥⌘–F), gathers together a number of powerful project-wide search tools for otherwise difficult to locate things, such as italicised text; cross-links; annotations & footnotes; and so on. The basic panel has a drop-down menu at the top which is the primary formatting mode selector. Below that there are two common tools to all modes.

Containing text Limits the search to only return results in which the specified text falls within the formatted range.

Search in There are two choices available: "All Documents" will search all text documents in the binder (even outside of the Draft); "Selected Documents" will constrict the search to only those items you have currently selected in the sidebar or a view.

If you leave all of the criteria types empty, it is possible to use this panel to find *all* instances of the particular formatting type.

The rest of the panel will change depending upon the current formatting mode. As with the standard text Find panel, it can be left floating over the project window as you click the **Next** and **Previous** buttons, allowing you to edit immediately after coming across a match, and then going back to search without having to call up the palette again. Also like the standard Find panel it can run in "background mode" as well. Once a search criteria has been set up, you can close the window and use menu commands or keyboard shortcuts for stepping back and forth from one match to the next, in the **Edit ▸ Find ▸** sub-menu.

The rest of this section will go over each of the individual formatting search types.

Highlighted Text

Looks for text that has been highlighted using the highlight feature (section 18.4).

Limit search to color When the checkbox is enabled, only those highlights matching the chosen colour will be found. If this is disabled, all highlight colours will be considered a match. Note that the colour must be *precisely* the same, so stick to using basic or custom swatches or the built-in highlighter defaults when using this tool.

Click once on the colour chip to bring up the colour palette, or right-click to select a built-in highlighter colour.

Comments and Footnotes

Will search within any inspector-based comments or footnotes in the project. By default all will be considered as potential matches, but you can narrow this down by selecting one of Comments or Footnotes from the "Type" menu.

Inline Annotations and Footnotes

Searching for inline annotations gives you three colour matching options:

- *Any Color*: No limits will be made on the search results.

- *Limit Search to Color*: will only consider annotations of precisely the specified colour as matching.
- *Exclude Color from Search*: any annotations precisely matching the provided colour will be excluded from the search. The default configuration of this search mode is set to exclude all "Blueberry Blue" annotations. This is because Scrivener uses this colour to add bookmark annotations. Since it is unlikely that you will want to find these types of annotations in your search, it is provided as a handy default.

Inline footnote searching is much more simple. Since footnotes cannot have custom colours, no additional criteria is necessary.

Revision Colour

You will be given the choice to search for a particular revision level in the drop-down menu. These are hard-coded as provided by the **Format ▸ Revision Mode ▸** sub-menu. Do note that if you have changed Scrivener's default colours, and are working with a colleague that has *not* changed their default colours, this can lead to situations where their revisions are not visible to you and vice versa. When working in collaboration with other Scrivener users, it is best to not customise these colours, or to share the list of colours used, prior to making any edits.

This feature will step through any edits that have been made while using a particular revision level, including any overstrikes that have been made.

Coloured Text

The interface for this type of search is similar to the highlight search type. You can provide a custom colour restriction in the additional criteria section. Note that since revision markings are essentially just formalised coloured text, you can use this panel to work around the problems brought up above, by searching for your colleagues custom colours manually. Do note the colour choice still needs to be precise, so using custom swatches or built-in presets will generally be easiest. As with the highlight tool, you can right-click on the colour swatch to select from a built-in default.

Links

You may search for URL or internal Scrivener Links using this tool. By default all links will be returned, but if you wish to narrow the search down to a particular link type, use the "Link Type" menu to make this choice.

While a link is selected, which will be automatically done for you when using the Next and Previous buttons, the **Replace with Title** button can be used to replace the visible text

of the link with the Binder title of the item it points to. This can be useful for updating cross-references in your draft, if you know the underlying name of the item has changed.

This can also be accomplished with the **Edit ► Update Links to Use Target Titles** menu command, which can be used to spot-fix links you come across manually, or to fix bulk portions of text in a single shot.

Text with Preserved Style

This will find text which has had a Preserve Style block drawn around it, from the **Format ► Formatting ► Preserve Formatting** tool. Since there are no optional qualities to these blocks of text, no additional criteria is required.

Character Format

Common text-level formatting can be searched for using this tool. Any number of options in the additional criteria can be stipulated. They work in an additive fashion, so if you have both Bold and Underline selected, a successful match must be *both* bold and underlined. "Keep with next" search for paragraphs that have the **Format ► Text ► Keep With Next** marker added.

21.2 QuickReference

QuickReference panels give you the ability to open more than two documents at once. They are in a way similar to Mac OS X's Quick Look, but rather than limiting you to only one Quick Look panel and a view-only model, many QuickReference panels can be opened simultaneously, and when viewing items which would otherwise be editable in the standard editor, allow full editing capability. They also provide full editing access to most data you can get in the Inspector, with the notable exception of snapshots (more on that below). Once opened, QuickReference panels will remember their size and position on the screen, as well as their meta-data viewing options. When "Reopen QuickReference panels when opening projects" is enabled in the General preferences tab (subsection B.2.1), any open panels, their position, and internal meta-data settings will be remembered in between sessions. Panels that were closed will not be remembered, nor will their various settings.

There are two appearance styles available. The default uses a standard window design. The alternate style uses a glossy black translucent style, familiar from Apple programs and Quick Look on Snow Leopard and lower. Both are functionally identical to one another, leaving this as a mainly aesthetic choice. The main exception to this is that standard appearance QuickReference panels can also use a Format Bar (which is hidden by default, but can enabled as per usual with the **Format ► Show Format Bar** menu command).

In addition, since these use a standard window, they will respond to normal windowing commands. You'll be able to minimise QuickReference panels to the dock and alternate through them with 'Cmd-". This choice can be made in the Appearance preference tab (section B.3).

An additional option, **Window ► Float QuickReference Panels** will toggle whether or not these should "float" over your other project windows. Floating is most useful in Composition Mode, where the backdrop would ordinarily hide the panels, but toggling this feature on and off can be useful in a variety of conditions. So a hotkey has been supplied as well, **^⌘-Q**.

The **Window ► Zoom** menu command, when activated on a QuickReference panel that is viewing an image resource will expand the size of the panel to match the width of the graphic, if necessary.

There are four (five, in OS X 10.8) easy ways to open a document in QuickReference mode:

1. Select one or more documents in the sidebar or a view and tap the **Spacebar**. If more than one document is selection, multiple QuickReference panels will be opened at once

2. Use the **View ► QuickReference ►** sub-menu to navigate your project binder and open a document from anywhere, without disturbing your existing editor views or requiring an editor lock. This method is also accessible from composition mode. This menu functions in a similar manner to the "Go To" menu. Each item in the Binder will have an entry, and you select a container entry just as easily, to open that container in its own QuickReference panel

3. Hold down the Option key when selecting an item from the **Go To** menu, or from the **Path** menu in the editor header bar. This works at any time

4. Drag and drop the item from any binder sidebar or group view mode onto the QuickRef toolbar icon.

Mountain Lion) Only Use the system standard three-finger tap to open anything in a QuickReference window. This is similar to a three-finger tap in Finder, when using Quick Look. This gesture works in the References list, Synopsis Finder, Binder sidebar and any group view.

The panel itself is composed of four major parts, one which is hidden by default. At the top you have quick access to the Label and Status meta-data fields. In the middle is the editor/viewer. This is an editable, full-power text editor when viewing standard

file or folder documents, and a standard media viewer just like the main editor when viewing PDFs, images, or QuickTime documents[1]. Below the editor/viewing area you will find a compact footer bar which will display a word and character count for standard file or folder documents. Beside that, a drop-down menu will give you access to most if document meta-data otherwise provided in the Inspector.

> **Text Zoom**: You can independently zoom the text scale within a QuickReference panel, either by using the keyboard shortcuts, ⇧⌘-, and ⇧⌘-. or by using the View ► Zoom ► sub-menu.

You might want to take a moment to examine the Navigation pane in preferences (section B.6) to see how you can use QuickReference panels to open things you might otherwise have thought you could only open in standard editor splits, such as clicked links.

Once you have closed a QuickReference panel, for the remainder of your session it will be saved into the Window ► Closed Panels ► sub-menu, much like a modern web browser might save all of the tabs you have recently closed. This way, you can feel free to close QuickReference windows whenever you are immediately done with them, as they will remain easily accessible to you within that session.

See also:

- Inspector (chapter 19)
- Meta-Data Types (section 10.1)

21.2.1 QuickReference Mini-Inspector

Each QuickReference panel has the ability to load up key components of the inspector's meta-data into a split view. Using the drop-down menu, you can select which type of meta-data you wish to view:

1. Synopsis: the content of the index card
2. Picture: the representational graphic if one has been selected
3. Notes: standard document notes (you cannot access project notes from a Quick-Reference panel)

[1] Unlike loading film and audio in a split, the remove pause and resume commands will not work in QuickReference panes, as you can have many audio files open at once, and Scrivener would not know which one to trigger.

4. Keywords: document keywords; this table also serves as a drop-target for keyword drags from the Project Keywords window.
5. References: the list of document references; as with notes, project references cannot be viewed.
6. Custom Meta-data: the custom meta-data fields in a linear list of editable text fields.
7. Comments & Footnotes: this is the only view that will pop-up on its own if you add a comment or footnote while writing in the editor. This pane works identically to its inspector counter-part.

21.2.2 Using QuickReference Panels in Composition Mode

Since by default, QuickReference Panels float over the Scrivener interface, they will remain visible even when entering composition mode, making them useful for bringing reference material into this dedicated writing environment. Additionally, you can open new QuickReference panels while in composition, by accessing the **View ► QuickReference ►** sub-menu.

If you have floating turned off, QuickReference panels will act like other windows in that clicking anywhere in the composition interface will hide them. You can bring all QuickReference panels back to the front by turning floating back on, even if only temporarily.

21.3 Goals and Statistics Tracking Tools

At some point, most writers will need to get some idea of the progress of their work by checking the word, character or page count. There are several ways of doing this in Scrivener:

- To get word, character and page counts for the whole of the draft (that is, the contents of the Draft folder) or for any documents selected in the binder, use **Project ► Project Statistics**. To calculate the printed page count statistics, Scrivener internally generates a compile document, based upon your compile settings—so you need to note that if you have compile set up to print off synopses and nothing else, only the synopses will be counted. This can also mean that there will be a delay in displaying this panel with larger projects, and with projects over 100,000 words, you'll need to manually update the printed page count, to avoid lengthy waits whenever the tool is used.
- To get word, character, paragraph and line counts for the current document, and a breakdown of word frequency (the number of times you use different words),

select the **Project ► Text Statistics** menu command (this will only be available if a text document has the keyboard focus). When using Scrivenings mode, the full word count will be tabulated, not just the section you are currently editing.

- To set a target word or character count and track your progress for the entire draft or the current writing session, select **Project ► Show (Hide) Project Targets**.
- To set a target word or character count and track your progress for a single document, click on the target button in the bottom-right of the footer view (this is not available in scriptwriting mode, though, where word and character counts are rarely useful anyway). See information on the footer bar for more information.
- Tracking a *group* of documents together as a cohesive unit, like say a sequence of sections and sub-sections within a folder, where the important detail is the word count of the entire folder is possible. You can use the Outliner to view the Total Progress and Total Goal columns, to aggregate statistics for group of documents. Setting a target on the chapter folder will, even if it has no text itself, work as an aggregate in Scrivenings mode, since child items will be included in the totals. Thus all words written to any subdocuments of it will count toward that aggregate goal, even if they themselves do not have individual goals. Targets for folders can be easily set by using the Target column in the Outliner, or by selecting the folder in the Binder, switching off the current group view mode, and using the target button as per normal in the text editor footer bar.
- To get the word or character count for a selection of text, select some text in a document and examine the footer bar of the editor. The word and character count will appear in blue text. When there is no selection, this area will be used to count the entire document or Scrivenings session.
- To view the word, character counts and targets for several documents at once, use the outliner view and make visible the appropriate columns by clicking on the "►►" button in the outliner column bar (or by selecting **View ► Outliner Columns**). You can use the "Total Words" column to collect aggregate counts for items which have children.
- To view a combined word and character count of an arbitrary selection of documents, use the outliner or corkboard to select several documents, and then Ctrl-click on the selection. The combined word and character count for those documents will appear greyed-out at the bottom of the contextual menu. Note this method only counts texts within the actual selection, not the implied selection in the case of children beneath the selected items. To count them too, disclose the items in the outliner and select them along with their parent items.

> **Usage Tip**: The word frequency tool in Text Statistics only counts text within a current view, which can include a Scrivenings session. Thus, to perform this analysis on the entire Draft or large portions of it, create a Scrivenings session first, and then use the **Project ► Text Statistics** tool.

21.3.1 Project Targets

The project targets panel allows you to set goals for your writing—either for the Draft as a whole or for the number of words you want to write during the current session—and to check on your progress. To bring it up, select **Project ► Show Project Targets** or press ⇧⌘–**T** to toggle its display.

The targets panel displays two progress bars: one showing the progress of the draft and the other showing the progress of the current session, , which you can optionally instruct Scrivener to reset for you at the end of every day. Using this panel, you can set a target word count for the draft (that is, the contents of the Draft folder) and for the current session, and keep track of how far you have to go before you reach your goals. To change the targets, just click in the appropriate text field and enter a new target word count (or target character count—you can set whether the target should be in words or characters by clicking on "words" so that a pop-up menu appears, providing a choice between "pages", "words" and "characters").

This panel will float over your project if you leave it open, and update itself in real-time as you write and edit. If you want to track a goal for an individual document, rather than the entire project, use the Text Goals (subsection 14.2.3) tool in the footer bar.

> **Using targets to hit an editing goal**: While this tool was originally designed to set a writing goal you work up toward, it can easily be used as an editing tool when cutting text. It not only counts words added, but words subtracted. You can set an editing goal in the same way you would set a writing goal. The progress bars will appear maxed out. Once you have edited below the target, you can be optionally alerted to this fact with Growl (or Notification Center on 10.8), or just keep an eye on it as you work.

If you delete lots of text, session statistics will not start showing any progress until you have written as much again—it is perfectly possible to have a negative session word or character count! In other words, it shows your net gain during the session. The session target only counts anything typed or pasted into a main text view (either of the editors or the composition mode editor) it does not count imported documents or appended text and so forth (so you can't cheat!). If you find yourself in a situation where you want to start fresh, because the counter is off, you can use the **Reset** button above the Session Target progress bar to reset the counter to zero.

[**10.8–9 Only**] In the Project Targets panel, you will see a Twitter icon. This accesses the universal 10.8 tweet tool. Scrivener will automatically generate a short sentence for you, based on any activity (or lack thereof) that has happened during the session. If you write 800 words, then it will select a phrase extolling how much you've written. If you've subtracted 800 words it will assume you have been editing that day. If there is no change, it will assume you have just started working for the day. By default it will use the @ScrivenerApp and #amwriting tags to help your tweets fit into the common network. You can edit the text that Scrivener supplies by changing it in the editor area. To use this feature, you must have a Twitter[2] account set up in your OS X System Preferences.

> **Not Seeing Twitter in Yosemite?** Because of changes in Yosemite, the Twitter and Facebook sharing services that were previously available in Scrivener will no longer be available when Scrivener is running on 10.10. These features are no longer available to 32-bit applications, and so will be restored in Scrivener when Scrivener goes 64-bit in a future version.

Project Targets Options

Click the **Options** button in the Project Targets panel to configure how these progress bars are calculated, and what Scrivener should do if you meet your goals.

Draft Target Options for total draft counting.

- *Count documents included in compile only*: will only monitor those texts which have the "Include in Compile" checkbox set. If your work style has note documents scattered about in the Draft folder, they will not be counted toward your final goal unless you disable this option
- *Target applies to the current compile group only*: takes this a step further and only calculates off of documents that have been chosen in the Content pane of the Compile interface. For performance reasons, this only works off of the top drop-down selection in the Content pane; dynamic filters will not be used
- *Deadline*: counts down the days you have left at the bottom of the target window. This can also be used to calculate how many words you need to continue writing per session in order to meet your deadline, below.

Session Reset Options There are four available options for how the session counter should behave. The counter can be reset every time a project is closed, you can have it track the time of day and reset it for you, or turn off automatic reset entirely and handle it manually with the reset button.

[2] http://www.twitter.com

- *Automatically reset at midnight*: when enabled, Scrivener will reset the session counter for you at midnight. For those with regular schedules who want to work toward a daily goal, this option is best.
- *Reset session counter on project close*: if you prefer to consider a session to be whenever you sit down and write, as opposed to a daily target, this option will be best for you.
- *Reset session count on next day opened*: with unusual schedules, such as those who write well after midnight, this alternate method of reset may work better. It will check against the last time you opened the project, and if it the calendar day has incremented it will reset. Thus, if you open the project at 22:00 and work until 03:00, when you open the project later on that day at 13:00, it will reset since you last opened it on the prior day at 22:00. If you start another session that night, however, it will not reset, since the last time you opened it was at 13:00 on that same day.
- *Never automatically reset session counts*: Scrivener will never alter the session progress counter for you. You will need to use the **Reset** button to start a new session. This is the best mode if you tend to work sporadically all around the clock in short bursts, or tend to set aside a "session" for a few days to deal with other tasks.

Session Target Options for session based counting.

- *Count text written anywhere in the project*: turning this on will count anything you type or paste into the *any* document in the project binder, even if it is a character sheet or a grocery list. When this is off, documents set to be excluded from the compile will not be counted even if they are in the Draft
- *Allow negatives*: when disabled, the session counter will never drop below zero. Leave this on to get an accurate net total of your writing session. When disabled, deletions will still be counted, but only until the counter reaches zero, so some deletions would no longer be counted after that point, making it less accurate
- *Automatically calculate from draft deadline*: requires the "Deadline" option to be enabled above, and the counting method to be either Words or Characters. Counting by pages cannot be accurately calculated when using this mode. When a deadline has been set, you can have Scrivener handle the calculation required to meet your deadline given the amount of time left. For example, if you are at 85k words in a 100k draft and have 10 days left to finish, then Scrivener will set your daily session goal to 1,500 words per day. If you

come in under or over that goal, Scrivener will adjust the daily session target whenever you hit reset, or automatically if midnight reset is enabled.

- *Writing Days*: by default, all days are considered eligible for writing, in terms of calculating your daily goal. If you cannot write every single day of the week, simply click on the days you can write, and the feature will adjust the calculation so that you don't end up writing below the curve. If all days are deselected, the calculation assumes you can write on every day.
- *Allow writing on day of deadline*: the calculator typically will not schedule you to be writing on the actual deadline day. If you want the system to give you up until the last minute, check off this box.

Show target notifications using Growl: if you have Growl installed on your system, Scrivener will register itself for announcements, and use that system to post notices when you exceed a goal, or fall back under it. [**10.8 (Mountain Lion) Only**] If Growl is not installed, then the OS X Notification Center will be used instead, and the label adjusted accordingly.

Show Twitter button: [**10.8 (Mountain Lion) Only**] optionally disable the Twitter button on the main Project Targets panel.

21.3.2 Project Statistics

Project statistics can be called up at any time with **Project ► Project Statistics** or ⇧⌥⌘–**S** on the keyboard. This sheet has two sections to it:

- Draft as Compiled: everything included in the Draft folder (and by default, only the current compile group). Note that compile options which modify export content in any way can impact this counter. If your compile is set up to only export titles in an outline format, you'll get a pretty small word count. Likewise, Compile content selection filters, and the state of the Include/Exclude/All selection drop-down will impact the total. In general, consider what your document looks like when you compile: that is what will be counted.
- Selection: a count of everything you have currently selected in the Binder, including all of their children items as well (which can be optionally disabled).

Each of these sections have identical statistics available to them. The word and character counters are self-explanatory. The two page count estimates are based on different algorithms.

Pages by paperback uses an industry standard formula (for English language publishing) of taking the average number of words per page and multiplying it by the average number

of characters per word (five including a space, for six total), the product of which is then used to divide against the total character count of the project. By example, a book with 720,000 characters with an estimate set to 250 words per page will produce a result of $\frac{720000}{250 \cdot 6} = 480$. If you need to adjust the average number of characters per word, switch the counting mode to Characters and provide the product of the words-per-page multiplied by average word length. For example, if the average word length is 8 and the number of words per page is 200, you would enter 1600 into this box to produce a custom result.

Pages printed will be more accurate for your own printouts, as it will compile your draft in the background, using the specified formatting and other content settings, and then count the total pages resulting from that. Consequently, in large projects it may take a few moments for this window to display. Once you exceed 100,000 words, the Statistics sheet will no longer automatically paginate every time you open it. You will need to click the **Update Printed Counts** button to manually recalculate printed pages. With large projects, it could be advantageous to tune the paperback estimated count to your printed count, so that it can serve as an immediate general figure to work from.

Project Statistics Options

To access options for how project statistics are calculated, click the **Options** tab in the window.

Draft Statistics Options Controls how the Draft only section of the stats are calculated.

- *Count current compile group only*: only calculates off of documents that have been chosen in the Content pane of the Compile interface. This is the only compile-time option that can be disabled. All other compile options that restrict or modify output quantity will still be factored into the count.
- *Count footnotes*: Footnotes are by default included in the count. If your publishing environment demands these be considered separate, here is where you can disable them in the total count.

Selection Statistics Options Tuning options for how selected items should be counted in statistics.

- *Count all documents*: In this context, all documents means: regardless of whether the document meta-data flag, "Include in Compile" is on or off.
- *Count only documents marked for inclusion*: Only those documents that have "Include in Compile" checked will be counted.

- *Count only documents not marked for inclusion*: As above, only with the inverse logic.
- *Exclude comments and annotations*: All inspector comments and annotations will be disregarded by default. If you wish to count them too, un-check this option.
- *Exclude footnotes*: Footnotes are by default included in the count.
- *Count sub-documents*: When disabled, only the literal selection will be counted. With this option on, the selected items *and* all of their children will be counted, all the way down to the bottom of the outline.

Page Count Options Set the counting algorithm used to estimate paperback page counting. This figure will be divided from the total word count to produce an estimate. Read the prior section for details on how this works.

21.3.3 Text Statistics

The Text Statistics panel can be accessed with **Project ▸ Text Statistics** or ^⌥⌘–**S**. It is only available when the active editor is displaying text, and it will produce statistics for *all* of the text being displayed, even if that editor is currently displaying many documents in a Scrivenings session. The types of information it provides are more pertinent to the topology of the text document, as opposed to the end product (where page counts and paperback estimations come into play).

In addition to the simple counter statistics, you can flip down the Word Frequency chart to display a table of discovered words, how often they have been used numerically, and graphically how often in comparison with other words in the table. These columns can be sorted by clicking on their header fields.

21.4 Recording Audio and Photo Notes

It is possible to take quick audio or photographic notation, directly into the Binder, using the New Media File tools in the Project menu. These tools only allow for the capture of new media, they are not intended as editing tools. Once audio or graphics have been created using them, you may load these files in an external editor to make any adjustments to the files themselves. They will be in every respect ordinarily media files that you can export, manipulate, re-import, or use in other programs.

21.4.1 Recording Audio

To create a new audio file, you can at any time use the **Project ▸ New Media File ▸ New Audio Note...** menu command. A small recording window will appear. This can be set

aside and used as needed while you continue to work in the project or reference material from it.

When the window loads, you will see a read-out in the middle stating whether or not the audio device in your computer is ready to record. If it says "Ready", click the large round button. While recording, the status read-out will display "Recording…", and the round button will change to depict a pause icon. Clicking this button will temporarily pause recording so you can gather your thoughts or look up material for the next bit of dictation While paused, the status read-out will print how many minutes and seconds long the recording currently is.

At any time you can click the **Save** button to finish recording and save the file to the disk. Once you have saved the file, it cannot be appended to or edited in Scrivener, but you can open the file in an external editor to clean up the pauses and append to it if necessary. If when you click the **Save** button, the binder selection is located anywhere outside of the Draft folder, the media file will be created in that location, according to the normal rules. If a normal file is selected it will be created below the selection; if a container is selected it will be placed within that container as the last child.

The **Cancel** button will delete any recorded material and not save anything to the disk.

21.4.2 Taking Pictures

To take a picture using an iSight camera, or the camera included in your laptop, use the **Project ► New Media File ► New Picture from Camera...** command. The most recent image you have taken will be displayed by default, or if nothing has been taken recently, an empty window will appear. You can select from a number of previous images to display in this window as well. Click the camera button to turn on the camera and start the standard countdown. During the countdown you can position the camera and target to get things framed as well as possible. A shutter sound will be played when the shot is being taken, and on some Macs, the display will flash white to better illuminate the scene.

Once the shot has been taken, you can use the slider below the graphic to change the size, and click and drag the image around to move the graphic around within the cropping size you've chosen. The effects button, to the right of the camera button, can be used to apply one of the default Apple effects to the image.

Once the **Set** button is clicked, the file will be reversed to account for the camera's single lens action, so any reverse text in the shot will be corrected for. Click the **Cancel** button to dismiss the sheet and opt to not save the file. The image will be saved as the last image taken, so it can still be retrieved if needed by calling up the sheet again. However it will not be saved to the Recent Images list until the **Set** button has been clicked on it.

21.5 The Name Generator

Scrivener comes with an exhaustive name generator, which includes many thousands of common names, as well as selections such as dictionary words that sounds like names, literary names from classics, a broad selection of regional names in several languages, and can even attempt alliterative names—all with extensive options for setting relative obscurity, naming styles such as double-barrelled surnames, initials, and so on.

To access the name generator, invoke the **Edit ► Writing Tools ► Name Generator...** menu command. You can adjust how many names will be generated, and then click the **Generate Names** button. Names can be selected for copy and paste, or if you right-click on a selected name, you can easily perform Google, Wikipedia or Spotlight searches, to check for existing uses of it.

To set further options, click the little gear button beside it. The available options are:

- *Gender*: select one option here. The default is either gender.
- *Attempt alliteration*: the generator will attempt to produce names with an alliterative effect, like "Jeromy Jin". This option will work best with Latin based languages.
- *Double-barrelled surnames*: produces names like, "Otis Cowie-Milburn"
- *Forenames use initials only*: reduces the forename to an initial. If more than one forename has been selected in the option below, multiple initials will be generated, like, "N. J. Pettersen".
- *Number of forenames/initials*: produces multiple forenames. You can select from 1 (default) to 3.
- *Set or search forename and surname*: this is a multi-purpose tool where you can either set part of the name yourself, or effectively search the database for a name by providing a part of it. For each part of the name:
  - Set: forces the respective name to be what you type into the text field; useful if you already have a name in mind but are having troubles with the other half of it.
  - Starts with: will only return names from the database that start with the characters you type into the field
  - Ends with: likewise, but for characters at the end of the name
  - Contains: the characters you supply will be searched for from within any portion of the name

- *Obscurity Level*: this slider to adjust how obscure the names should be. Moving the slider all the way to the left might produce a result like, "Scott Young", while sliding it all the way to the right, "Chauncey Noach" (no offense to all the Chaunceys out there). Note that some lists by definition are obscure, so if you keep getting odd names, check your source list and make sure nothing unusual is selected.

Finally, you can select from one of the many lists provided. You will always need at least one Surname list selected (indicated by the yellow "S" icon), and at least one gender list which is compatible with the gender option set above. If you have the list set to return male names, you will need at least one male list of forenames selected. These are indicated by the standard gender symbol icons. You can choose as many sources lists as you like at once.

The built-in lists will be shaded in grey because they cannot be deleted.

You can add your own custom name lists to the generator. Name lists should be formatted so that all names are on a single line, and each name is separated by a comma, like so:

```
name1,name2,name3,name4,name5
```

It might be easiest to produce these lists in a spreadsheet on a single row, and export as a CSV file. If you use a regular text editor, make sure to name the file with a ".csv" extension, and then click the + button in the Name Generator option panel. Locate the file in the chooser dialogue and click **Open**. You will be presented with some options on how the list should be marked. Give it a descriptive name, and mark whether it is a list of female names, male names, or surnames. Finally, if you have ordered list from most common to most obscure (at the end of the list), check this box to enable the Obscurity Level slider. Note that in small lists, the obscurity slider may not have much impact, depending upon how many names are being generated.

If you wish to delete one of your custom lists, select it in the list and click the - button.

21.6 Bibliography Management

Scrivener offers simple integration with your favourite bibliography or citation manager (such as EndNote, Bookends or Sente) for academic work. To set it up, visit the General preferences tab (subsection B.2.6). Click on **Choose** to choose your citation manager (usually from the Applications folder).

Once the citation manager has been set up, you can hit ⌘–Y (or **Format ▶ Bibliography \ ▶ Citations...**) while working in the editor, to bring your chosen software to the front, launching it automatically if necessary. The steps you take next will depend upon your

citation manager, so you will need to consult their documentation on how to use their software with third-party word processors. Typically, you would copy and paste a citation placeholder into Scrivener, in the location where the reference mark should appear. After compiling to RTF, you would then use your manager to scan these placeholders into final print form. Not every program or service provides RTF scanning, however. You should research the software you intend to use and make certain it designed to work with programs other than Microsoft Word.

> **Using Scrivener with Papers**: If you are using Papers version 2, which has citation management features, it is better to use their built in "Manuscript" feature than Scrivener's integration, as this will provide a quick pop-up interface that automatically pastes the citation placeholder back into Scrivener. Refer to the documentation for Papers for details on how to use this. Papers works best with the .doc or .docx format rather than RTF, which means you should use the improved document converters in Scrivener for best results (subsection 24.3.2).

21.7 Using Equations with MathType

If part of your writing involves the addition or construction of equations in your draft, then you can use Scrivener's Design Science MathType[3] integration to insert equation objects into the draft, much like you would an ordinary figure.

To create a new equation, position your cursor where you wish to have it appear, and use **Edit ► Insert ► MathType Equation ►**. If you have MathType correctly installed on your machine, you will see the equation interface pop up over the Scrivener window. Any changes made within this window will be saved back into the Scrivener project when you close the window (by default it will ask for confirmation when closing the window) or use **File ► Close and Return to Scrivener**.

To edit an equation later on, simply double-click on the equation in the Scrivener editor. The MathType interface will pop up again, and any changes you make will be saved back into the file when you close it.

Equations act much like ordinary images. They can be aligned or styled in the editor, and when they are compiled they will be converted to images and handled as they ordinarily would be for the particular format in use.

Equations can be saved and edited from any rich text field, including project or document notes.

[3] http://www.dessci.com/en/products/mathtype/

Typing Expansion Conflict Some type expansion applications, such as Typinator and TextExpander can cause Scrivener and MathType to have troubles communicating with one another. The main symptom of this will be items that refuse to save changes, or the inability to insert new equations. If you experience either of these symptoms, restart Scrivener and avoid using any text expansion tools while working with equations (most programs give you a setting to ignore particular applications, you may wish to take this step).

Chapter 22

Using MultiMarkdown

For those who prefer structural writing solutions to rich text (and if this means nothing to you, you can entirely skip this section), Scrivener allows you to import and export using Fletcher T. Penney's MultiMarkdown syntax. MultiMarkdown makes it easy to generate documents in any number of formats—for instance, LaTeX and HTML5 - using a basic markup syntax (for instance, using asterisks to define **bold** and *italics*).

It is important to note that if you have used normal word processors (such as Microsoft Word or OpenOffice.org) and rich text for years and have never heard of MultiMarkdown, structural writing or such formats as LaTeX, you can ignore this aspect of Scrivener. Scrivener was designed primarily as a rich text application; MultiMarkdown is implemented in such a way that those who want it can access it easily but those who have no need of it need never be bothered by it. If you are curious about the process and would like to consider adapting it, you should read What is MultiMarkdown? (section 22.2) for an overview of its philosophy, limitations and capabilities.

22.1 What is Markdown?

The MultiMarkdown syntax is based on the "Markdown" syntax created by John Gruber, and the best description of what Markdown is comes from his Markdown web site, Daring Fireball[1]:

> Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).
>
> Thus, 'Markdown' is two things: (1) a plain text formatting syntax; and (2) a software tool, written in Perl, that converts the plain text formatting to HTML. See the Syntax page for details pertaining to Markdown's formatting syntax. You can try it out, right now, using the online Dingus.
>
> The overriding design goal for Markdown's formatting syntax is to make it as readable as possible. The idea is that a Markdown-formatted document

[1] http://daringfireball.net/projects/markdown/

> should be publishable as-is, as plain text, without looking like it has been marked up with tags or formatting instructions. While Markdown's syntax has been influenced by several existing text-to-HTML filters, the single biggest source of inspiration for Markdown's syntax is the format of plain text email.

22.2 What is MultiMarkdown?

For comprehensive information about MultiMarkdown and how to use it, see Fletcher T. Penney's web site[2]. The purpose of MultiMarkdown is to extend Markdown's goal (which is primarily toward web publishing) into the realms of traditional publishing, while still retaining a solid platform for web publishing, and adhering to the technical and aesthetic principles of Markdown. Toward this goal, it provides output formats for OpenOffice's ODT format[3], simplified RTF files and the LaTeX typesetting engine.

To augment the expansion into document formats, additional features have been added which will be of interest to authors, such as footnotes, tables, and cross-referencing.

The basics aside, you might be wondering if any of this is relevant to you. With the exception of generating LaTeX documents, it should be made clear that there is nothing you can do with MultiMarkdown that you cannot do using Scrivener like a typical word processor. That's a very broad statement, specifically there are many differences, but most of these revolve around how you work rather than what your end product of that work will be. We'll get to the specifics shortly, but for now here is a list of differences in the end product that you can expect:

- MultiMarkdown (hereafter shortened to MMD) can produce *styled* word processor documents, when using ODT. That means your document titles will be classed according to their depth in the outline, block quotes will be assigned to the appropriate style, and so forth. If you require a stylesheet-ready word processor document at the conclusion of the writing process, MMD is something to consider.

- As already stated, MMD supports easy to write LaTeX document generation. Without having to bother with the nitty gritty codes, you can produce clean, attractive documents using a typesetting system that has been in refinement for nearly thirty years. How good are these results? Well you're look at an example right now. The Scrivener user manual is typeset using this system. If you require LaTeX documents,

[2] http://fletcherpenney.net/multimarkdown/

[3] via the Flat XML format (.fodt) which is currently best supported by the OpenOffice version, LibreOffice.

but would rather not *compose*LaTeX documents, MMD might be a good choice for you.

- For web publishing of all kinds, MMD outputs clean and semantically valid HTML5. What does that mean? It means your titles will be defined into outline logical header depths, your quotes will be blockquotes, your tables will be clean, and most importantly the output will be easy to apply CSS stylesheets to. This will produce a result you can often just drop into your blogging software without any stripping of codes or fiddling with styles.

- Currently the best tool for turning a MultiMarkdown document into an ePub book is to use the Pandoc[4] document conversion engine. From Scrivener, compile a plain MMD document to use as the source for this conversion.

Now for the specifics, or what we might call the 'workflow'. One of the thing that attracts people to MMD is that it focusses solely upon the text. All formatting is done with easy-to-read text codes. For the most part, these are also very easy to type in. You don't have to memorise keyboard shortcuts for formatting commands because they all use common punctuation marks.

On that note, you're going to have a stare at a little more punctuation than you might be used to. This isn't as bad as writing in HTML, not by a long shot, but there is very little rich text to MMD conversion in Scrivener. You won't be looking at normal italicised fonts and such. Some people welcome that, others are glad that 1995 happened and never want to see codes again. If you'd like to see just how much punctuation we're talking about, you can find the Scrivener user manual project on our support page[5]. Download that and browse through the project. Keep in mind that, as a technical document, it has a more dense formatting requirement than a novel or non-fiction document might. Even so, you should find that it doesn't look all that much different from a standard document.

This chapter will not attempt to teach you MultiMarkdown. There are excellent resources available on the Internet for this, including the link at the top of this section. If you already know Markdown, then you already pretty much know how to write in MultiMarkdown, especially with Scrivener, which handles some of the MMD extensions for you. It will instead cover the integration features provided by Scrivener, and assumes a basic working knowledge of the system.

[4] http://johnmacfarlane.net/pandoc/
[5] http://www.literatureandlatte.com/support.php

22.3 Getting Started with MMD

There is very little you need to do to prepare for using MultiMarkdown. Scrivener comes pre-loaded with the latest version of it (at the time of its release), and also includes several optional packages which will be employed as needed. This means that even if you intend to use a more advanced workflow, like LaTeX or XSLT post-processing, you won't need to install anything extra. You are of course free to, and Scrivener will cross-check your system for you. If it detects installed components, it will gracefully switch to using them behind the scenes. You could right now start testing it out and compiling finished documents.

If you've already got a Markdown or MultiMarkdown document in a partial state of composition, you might like to know you can import it into Scrivener. If the document does not have titles in it, then this method will be of no use to you, you might as well just drag and drop the file straight into the binder to import it. If it does have titles, then use the **File ► Import ► MultiMarkdown File...** menu command. This will read in the title structure and convert it to a Scrivener outline, parcelling out all of the text in between titles into the appropriate sections of the binder. As you can see, this makes for a good way of "round-tripping" a document if you need to pass it along to someone else for editing, or wish to edit it yourself in another piece of software.

22.4 MMD and Scrivener

Because MultiMarkdown requires a certain degree of textual precision, many of the visual tools that rich text users commonly use can have a negative impact on how well you can "see" your document. For example, in MMD, a tab in front of a line turns it into a code block; thus indenting the first line of each paragraph could lead to confusing results. Likewise, using paragraph spacing could make for confusing results in an environment where every block needs a clear space around it. Pseudo-spacing might lead you to believe a space exists where in reality there is none. Consequently it is often a good idea to reduce the level of formatting in your projects as much as possible.

This doesn't mean that a project needs to look unappealing. If you reduce the paragraph spacing to zero, the double-newline that is required between paragraphs looks only slightly larger than normal. Meanwhile, because formatting is ignored by the compile process, you can feel free to take advantage of any formatting tools you like to pretty up your document, safe with the knowledge that none of it will ever appear outside of Scrivener. Some like to use a monospace font, like in many plain-text editors. Indeed, with tables, this can make editing much easier. However many others use whatever font is appealing to them, maybe saving monospaced fonts for those sections that require a little perfect placement.

What about special characters? This is often a matter of taste and workflow, but you should be aware that many of the default outputs do not handle Unicode well, and it is often better to either use codes for special characters, or let MMD handle the more common ones, such as typographic quotation marks. You will often find that MMD works in a more error-free fashion if you disable all of Scrivener's Corrections (section B.9) preferences. You won't need to worry about which way your quotation marks are pointing, just leave them straight and let MMD handle the typesetting details for you.

There is one exception to that. An option in the compiler can be set to attempt to convert rich text formatting to real plain-text whitespace. That is, a paragraph with a full line of pseudo-spacing after it will be converted to an actual empty newline. Indents will become tabs (so be careful with that one). This option is mainly intended to be a convenience for people who are switching to MMD from a rich text environment, and have a lot of existing material that is single-line spaced. However you could use this option to take advantage of rich text aesthetics while working in MMD, if you wished.

When exporting via MultiMarkdown using any of the above methods, all rich text formatting is completely cleaned from your draft, as it passes through a plain-text engine; there is no need to clean up extraneous highlighting and other markings when working with MMD. There are a few exceptions:

1. Any inline or linked images will be converted to MMD syntax, and exported into the compile folder, ready to use by the HTML, Markdown plain, and LaTeX formats. Any embedded, linked or placeholder style images will be converted to MMD syntax and exported into the compile folder.

2. Images that have been linked to, using Scrivener Links, will have the linked text replaced with the image name, and the image itself exported to the compile folder. Thus you can type custom image syntax into the document, and still benefit from automatic resource gathering. **Hint**: you can use Opt-Drag to drop a Scrivener Link to the image, as text into your editor, forgoing the need to type it in and then link it manually.

3. Scrivener's notation features (both inline and linked) can be used to generate MMD footnotes, and optionally its annotations and comments can be inserted into the document as well. Inline annotations on their own line (not embedded in another paragraph in any way) will be exported as DIVs, all other annotations, and all comments, will be exported as SPANs, to preserve the original paragraph flow around the comment.

Additionally, while Bold and italics cannot be compiled, but if you prefer to write using them, they can later be converted to MMD asterisks with the **Format ► Convert ► Bold**

and Italics to MultiMarkdown Syntax menu command. Since rich text formatting can be less precise, you should verify that the syntax produced by this command is valid.

A few of Scrivener's tools, intended primarily for a rich text workflow, have alternate purposes with MultiMarkdown:

Document Titles in the Compiler Document titles will be converted to MMD syntax. While this is rather a function of the compiler (which may be set to not include titles) rather than the editor, it is important to know that your outline structure can still be seamlessly impressed upon the final work as output structure.

When titles are set use run-in formatting so that they appear in the same line as the first paragraph of the text, hashmark generation will be disabled.

Preserve Formatting blocks Accessed from the Format ▸ Formatting ▸ Preserve Formatting menu, these blocks can be applied to text to protect them from formatting alternations during compile. With MMD, since there are no formatting alterations, these blocks will instead emit a single tab character in front of every paragraph within the block, in effect, making it a code block or a poetry block (depending on which exporter you are using).

Compile As-Is This checkbox in the inspector is generally used as a larger area of effect version of the above tool: it will protect an entire document from formatting alterations. In MMD, all this checkbox does is restrict the document from emitting a title header if it otherwise would have. Thus it can be useful for injecting sections of pure LaTeX, use custom header depths that otherwise violate the outline's literal depth, and so on.

22.4.1 Images

As mentioned before briefly, images can be dropped into the editor where you would like them to appear[6], just as you would do so in a rich text work flow.

Also as with rich text, if you resize the graphic in the editor, instructions will be sent to the display formats to size the image in accordance with your specifications. This will change the images display dimensions in its meta-data, rather than resampling the image,

[6] Though do note that the LaTeX typesetting engine rarely takes such placements as literal. It will place the image (and tables for that matter) as close to the point you specify as possible, while maintaining a aesthetic appeal to the text blocks around it. For instance, if the placement of an image would push three words of the last paragraph onto the next page, it would prefer to keep the paragraph intact, and move the image to the next page instead.

and provide that dimension data to MultiMarkdown using its syntax for declaring image sizes.

There are cases where you might not want the image to be in the editor. The image might be too large to comfortably work around, being print resolution, or you might find that the inclusion of many graphics in your files slows Scrivener's saving routine down. While you could use the image placeholder tag that is available to rich text authors, you might as well use MultiMarkdown's image syntax for its expanded feature set. To do so, simply write the image syntax as your normally would, and where you need the image's path to be printed, create a Scrivener Link pointing to that image in the binder. It is important to note that any text in the hyperlink itself will be altered to the image's file name, so you needn't even be aware of the image's name to use this method, but will need to avoid linking to graphics in such a way that the hyperlink extends beyond the area of the syntax where the path name should be printed.

Cross-referencing Images

The manner in which Scrivener generates image syntax is conducive to cross-referencing to them. It will use the image filename for the ID. All figures with a generated ID will be placed in a list at the bottom of the compiled document for easy reference, along with any footnotes. Images that do not require an ID (such as inline paragraph images) will not be handled this way and cannot be linked to[7]. For images to have a generated ID, they must be placed in their own paragraph with no other text (excluding captions, described in the next section).

In example, if you drag a graphic called 'analysis_of_derivatives-2008.png' into the editor, Scrivener will export that graphic into the compile folder (subsection 22.5.1) and generate syntax where the position was located, like so:

```
![][analysis_of_derivatives-2008.png]
```

Then placing a the data for the graphic at the bottom of the compiled document:

```
[analysis_of_derivatives-2008.png]: analysis_of_derivatives-2008.png
```

You can thus refer to this figure in another portion of the document with the standard MMD syntax, **(see chart)(#analysis_of_derivatives-2008.png)**.

[7] However if an inline images has been resized, it will used the referenced syntax to apply the dimensions code to the image.

Captions

When images fall upon their own line, you can supply a caption to them using one of the following techniques:

- Place the caption within straight-double-quotes on the same line after the image, separating the image from the quote with a space
- Using the same method as above, replace the double-quotes with matched square brackets
- Or you can use square brackets on the line *directly* following the graphic.

In all cases, there must be one whitespace character between the image and the caption syntax (either a single space or a single carriage-return), and there can be no text following the second quote or close bracket, otherwise it will be interpreted as a normal paragraph and the image will be exported using non-referenced syntax.

Captions will be exported to all formats which have a way of recognising them. If the format does not recognise captions, then nothing will be shown. ODT, HTML, and LaTeX all support captioning. HTML5's figcaption syntax may not be visible in all browsers unless it is specially handled by CSS.

22.4.2 Footnotes

You needn't worry about footnote syntax and identification at all when using Scrivener to compose MultiMarkdown documents, though if you wish to, there is nothing stopping you from typing in the codes yourself. You can simply use Scrivener's two tools for marking text as footnotes. Whether they will ultimately become footnotes or endnotes is not dependent upon anything Scrivener can do, but upon the export format's limitations or options. Under the hood, Scrivener will automatically number your footnotes as they occur in the document, and build the proper MMD syntax for them.

It will use two different naming conventions, depending upon whether the footnote was inline or a linked inspector footnote:

```
[^fn1]: This came from an inline footnote.
[^cf1]: This came from an linked footnote.
```

This distinction, if going straight to one of the basic output formats, will be of no concern to you. However if you intend to post-process the results using your own scripts or stylesheets, this distinction will enable you to handle two streams of notes in your document. You could for instance use inline footnotes as footnotes, and linked as endnotes.

The placement of the marker in the source text will follow the same rules as standard Scrivener usage. Mentally replace the entire inline footnote with the marker, to see where it will be placed, or the trailing right-edge of a linked range.

22.4.3 Annotations and Comments

For most uses, inline annotations and linked comments will serve much the same purpose when using MMD as they would otherwise. They are a convenient way for you to apply notes to your document, or to share and receive thoughts from those who are editing your work.

Since MMD itself does not have any syntax for commenting a work (unless of course you consider HTML comments, which reduces the flexibility of your document as these are otherwise used to pass raw LaTeX code to the final output), Scrivener provides two choices to you for handling them:

1. Inline annotations can, as with the other plain-text output formats, be wrapped in enclosing markers (square brackets by default). Creative use of this can allow you to use the inline annotations as shorthand for a longer sequence of coding you typically use—for example you could use them to supply HTML comment markers around the annotated text, thus providing an easy to read way of pushing raw LaTeX code into the final compile.

2. Inline annotations and linked comments can optionally be compiled using HTML syntax, instead.

When converted to HTML, linked comments and inline footnotes which are in one way or another attached to another paragraph will be converted to elements, which have been classed as "annotation", and coloured with inline CSS in accordance with their appearance in the editor or sidebar. If however an inline footnote falls on its own paragraph, it will use a element instead. While simple MMD syntax can be applied to a span range (such as italicised text), a div block can contain complex MMD syntax, and thus an inline footnote can feasibly support all of the things that normal text in the editor could, such as images, lists, tables and so forth. The only thing you cannot do within such a block of text is footnotes, since footnotes and annotations cannot overlap in Scrivener. When a span style annotation has paragraph breaks within it, Scrivener will inject <br/> elements to emulate paragraphs without causing syntax errors. All of this means that the original flow of the document, with or without notation, will be preserved.

> **Using Annotations as Functional Ranges**: Furthermore, and especially in conjunction with post-processing systems like XSL, the usage of HTML elements for notation allows for some creative post-processing uses, effectively allowing you to expand the syntax of MMD as needed, or even just as a way to better style your document with CSS. In fact the Scrivener user manual that you are reading right now makes heavy use of this technique. Whenever you see a tip box, such as this one, you are looking at what started out as an inline annotation in Scrivener. Ranges of text which pertain to only one platform or another (such as keyboard shortcuts) are blocked out with annotations as well, so that only one type appears. There are numerous strategies for doing so, but it is not within the remit of this manual to get into detailed examples of post-processing. Visit the MMD section of our user forums[8] for archived discussions on the topic, or start one of your own.

22.4.4 Preserve Formatting

The Preserve Formatting (subsection 15.4.6) feature in Scrivener is typically used to protect a range of text from being reformatted in the compiler. Since formatting in MMD is discarded anyway, this tool has been co-opted to produce code blocks (or pre-formatted ranges). When used within a paragraph, they will wrap the preserved text in ‘back-ticks‘, which will inhibit the interpretation of any processing within the range. You would be free to use MMD coding within the Preserve Formatting block without it turning into formatting. In default conditions, this will also cause the text to be displayed in a fixed-width font.

If the Preserve Formatting block is on its own paragraph, then Scrivener will switch to using the multi-line code block syntax, which is simply to insert a tab in front of each line that falls within the block. This is thus useful for presenting programming language code and other larger scale sequences of text that should not be interpreted for formatting, and have their whitespace handled verbatim.

Some alternate options (not supplied by default in Scrivener, but a part of the MMD eco system) can format code blocks such as these as verse instead of “code”, making it useful for poetry and other forms of text that require custom spacing within lines.

22.5 Compiling

This section of the documentation will not cover compilation in detail. You should consult the chapter on Compiling the Draft (chapter 24) for detailed information on the various options that will pertain to you. Most of the MMD options are built into the existing option panes, often replacing those that are not relevant, or changing options

as necessary. This section will instead focus on the various formats available, and which compile panes to check for best harnessing Scrivener's compile automation.

> **Using MultiMarkdown 2**: If you have documents which still rely upon MMD2 syntax, or just prefer to use that system instead of the new one, you can still do so with Scrivener. You will need to use plain MultiMarkdown as your compile option (you can no longer go straight to a format, and will need to do the post-processing yourself). When this option is selected, check the "Use MultiMarkdown 2 compatibility" option the Compatibility (section 24.23) option pane. All generated syntax will be in compliance with MMD2 protocols.

22.5.1 Compile Folder

As with the standard HTML output in Scrivener, if the compile results in more than one file being produced, a compile *folder* will be created using the name you specified, with the output document located within that folder along with any support files it needs. This will most often be graphics, or supporting .tex documents for typesetting.

You may find that you want to keep this output folder "static" instead of having Scrivener completely regenerate it each time you compile. You might wish to assemble CSS files, custom .tex files, styles, and other bits of supporting data so that you can compile directly into a finished location, rather than compiling and moving things around into the final support folder each time. Scrivener will for the most part handle this transparently for you with natural file extensions. If the compile folder, for example, is called "my_novel.tex", then when you compile to that folder Scrivener will leave what is already there alone, unless it is a file that Scrivener is scheduled to produce. This way graphics will be updated, but any logos or other elements from the title page that are not in the project itself will be retained. Extension detection is done for ".tex", ".md", ".mmd", ".fodt" and ".html". If the file extension you wish to use is not one of those, you can instead rename the compile folder to "my_novel_mmd" and Scrivener will leave it alone the next time around. Any folder ending in "_mmd" or "-mmd" will be preserved.

> **The "Overwrite preserves other existing files" Checkbox**: In the compile file save dialogue box which appears after clicking the Compile button—for MMD formats which have the potential to produce folders instead of singular files—you will find a checkbox labelled "Overwrite preserves other existing files". This option accomplishes the same as above with one slightly different variation in that you use it to compile to the parent folder containing the compile folder, not to the file *within* the compile folder. Thus, if your "my_novel.tex" folder is in Documents, you would compile to Documents, opting to overwrite the folder. With the checkbox on, anything in that

> folder not otherwise slated to be produced by Scrivener will be preserved. With the checkbox disabled, the folder will be completely refreshed.
>
> Users of the **Mac App Store** version will need to use this feature exclusively, as Sandboxing does not allow the dynamic folder preservation technique.

22.5.2 Plain MultiMarkdown

While the built-in compile options that go directly to a specific format are incredibly handy, there are occasions when exporting a plain MMD document can be useful and even valuable. The MultiMarkdown format makes for an excellent long-term archival document. It's compact, easy to ready, and easy to transform into a variety of output formats. It's also very portable, being plain-text. While dropping an archived Scrivener project that you are done with into a database system or even just in your file system somewhere is something you should always do, a plain-text copy of the final product is something that will be more easily archived in multiple places for many years to come.

Beyond archival, a plain MMD document allows you to "step into" the compile process a bit, which can be useful if you intend to do advanced post-processing of your own, or are getting errors with a particular format and need to figure out what is causing the syntax to foul up. The "MultiMarkdown" selection in the Compile For menu is precisely the same copy that Scrivener will be using to feed MMD, when generated a direct formatted output, and so is a critical troubleshooting step when faced with difficulties.

It is also, as noted above, a necessary format if you intend to use the MultiMarkdown version 2 system, as there is no longer an automated way of doing so in Scrivener.

22.5.3 LaTeX

If you've used MMD in the past, you'll have probably noticed that how LaTeX documents are set up in MMD3 is quite a bit different than in the past. In the past you could just tell a minimally annotated document to be a .tex file and it would do so. If you didn't like the default Memoir class, you could throw in one line of meta-data to change it. This system, while easy to use from the document end, was rife with difficulties if you ever wished to slightly tweak the .tex output. You had to learn at least a little of the complex XSL stylesheet language to make your own class definitions.

The new system alleviates much of these problems by using .tex boilerplate files that you can easily edit in any plain-text or TeX editor. These are then imported into the final .tex file with the `\input{}` command, leaving the base document clean looking. This has however come at the expense of a slightly more involved meta-data block, and an awareness of how to install custom .tex include files into your system. We have implemented a system to reduce the necessary "plumbing" required to set up and switch between LaTeX

stylesheets, as well as handling the production of gathering of any necessary files for one-click typesetting. In fact, you may not have to touch a single setting to get a ready-to-typeset document.

> **Been Using MMD3 for Some Time?** When you select the "MultiMarkdown -> LaTeX" compile method, a "LaTeX" compile option pane will be added to the list. By default, your document will be compiled to the Memoir (book) class as it always has in Scrivener. If you wish to change that, or already have your project set up to define the class, you'll want to visit this pane and change the provided setting. If you don't do that, then end result could have doubled-up meta-data keys.

The automatic class selection system will insert the necessary meta-data keys where they should go, amongst any you have declared, and if necessary it will produce for you any of the boilerplate .tex files these classes depend upon. This will all be done during the compile phase, so the meta-data keys being used will not be displayed in the Meta-Data options pane. If you already have these files installed in your LaTeX distribution, then they will not be produced into the compile folder. If you would like to install these, you will need to download them from the MMD LaTeX support file repository[9], and install them into "~/Library/texmf/tex/latex/mmd/", or wherever your distribution handles user texmf files.

You can mingle "LaTeX Input" keys in your own meta-data, if they are meant to be used in conjunction with the defaults, so long as you place them so that they appear in the appropriate position upon final output. An example might be a custom chapter title typesetting rewrite that you'd prefer to use instead of the stock default.

In addition to the default classes, you can write your own in such a way that they are saved into the compile settings (or ultimately a preset) using the "LaTeX Options" compile pane, and selecting "Custom". Three text fields will be provided, where you can supply the initial preamble, and further preamble that is dependent upon any meta-data values (such as a title page) and beginning the document, and finally a footer placed at the bottom of the document. They will be exported into the compile folder and linked up in the generated meta-data for you. So long as the syntax is correct, your document should be ready to typeset moments later.

22.5.4 RTF (via textutil)

There is very basic support for the Rich Text Format. MMD will create an HTML file behind the scenes, and then use your Mac's *textutil* utility application to convert it to an RTF file. Since it goes through HTML to do this, feature support is limited mainly

[9] https://github.com/fletcher/peg-multimarkdown-latex-support

to "what you see", and images will be dropped from the output. This method is best employed when you want a quick proof that can be viewed in almost any word processor. If you have need for a more comprehensive word processor workflow, you should use the ODF method instead.

22.5.5 HTML

Produces a clean, semantic HTML document. In older versions of MMD, this used to be the base format upon which all other formats were derived (using post-processing scripts). It is now simply an parallel format on equal footing with the rest. It is however still used as a base format if you invoke the XSLT post-processing switch in the Compatibility option pane. The default output will be basic HTML. If you wish to supply CSS to the document, you can do so with the "CSS" meta-data key (specifying a file or URL), or type one in manually into the "HTML Header" key, which can also be used to add Javascript and other functionality to the document.

22.5.6 ODF (Flat ODT)

If you need your work to enter a standard word processor workflow, this will be your go-to method. MultiMarkdown's ODF support is comprehensive, supporting most data types, and will utilise stylesheets, meaning the document can be easily adjusted aesthetically, and use the table of contents feature of a word processor. It represents the only output method in Scrivener that offers this support, making MMD an enticing option for those who do not mind learning how to use it, if they need stylesheet driven word processor documents.

The OpenDocument Format is an open specification office format, and specifically the ODT format within this specification is for word processing. OpenOffice (and the many forks off of it) is one of the few word processors that fully addresses the format. MultiMarkdown uses the *flat XML* variant of ODT, which has the extension of .fodt. At the time of this writing, LibreOffice (an OpenOffice fork) offers the best support for .fodt files, though OpenOffice proper can be updated to handle it as well—in fact LibreOffice just has this update preconfigured. Once the file has been loaded in LibreOffice, it can be saved out to many other word processor formats, a task you will almost always need to do if working with others.

22.5.7 PDF (via LaTeX)

[**Standard Version**] The PDF output choice utilises the LaTeX typesetting engine, and simply offers a streamlined way of going from Scrivener to a printable PDF in a single step. It is fundamentally identical to compiling as a .tex document, and then opening that document in a TeX editor and typesetting it with no alterations (three times, to get all

cross-references properly registered). Consequently, if the meta-data calls for "snippet" mode instead of full document mode, are fatal errors in the typesetting process or if it requires additional steps to be completely typeset (glossaries and bibliographies foremost), it will not produce the desired result, if any.

If the document has no extra requirements, and typesets cleanly otherwise, you will find it to be a most convenient way of distributing and archiving great looking printable documents. When troubleshooting or performing initial typesetting it will often be beneficial to enable the "Show PDF Log" option in the Compatibility option pane. This will produce the full .tex log file, and so reveal any warnings that might otherwise be ignored.

Since this method combines a full compile with full MMD post-processing and three *pdflatex* executions, large documents may take a while to fully complete.

> **Where is it?** If you do not have a LaTeX distribution with *pdflatex* installed on your computer in an executable path, then this option will not appear in the compile format drop-down.

22.5.8 XSLT Post-processing

This option, located in the Compatibility pane, is available for all MMD output formats except plain MMD, ODF and RTF. You will most likely want to install your own copy of MMD3's Application Support package[10]. This will provide an XSLT infrastructure so you can make modifications to the workflow. Scrivener will use its own built-in copy if you do not have one, but you'll be limited in what you can do with it, since the XSLT files cannot be safely edited (each time you update Scrivener, your changes will be lost inside the application bundle).

In a nutshell, MMD will create an HTML5 document internally, and then use an XSLT script to transform the XML of this document. The extent of these modifications can be minor or radical. A simple example of a minor modification would be adding a small table of contents at the top of the web page. A more radical example would result in a plain-text file suitable for Wiki publishing, or even a LaTeX document. If you desire full control over how every single element that MMD produces is in the end created, then XSLT is what you want. Those adept in MultiMarkdown version 2 may also find this a more comfortable (if slower) way to work.

Please read the MultiMarkdown documentation for how to use this feature. In addition to enabling the option (which will only switch the post-processing scripts to the XSLT workflow) you will need to specify and create XSLT scripts using the proper MMD meta-data keys for doing so: "LaTeX XSLT" and "XHTML XSLT".

Should you use it? Here are some scenarios that might help you decide:

[10] https://github.com/fletcher/peg-multimarkdown/downloads

- If you want to change the document class to a special stylesheet you prefer in LaTeX, you don't need to use XSLT for this. You can either roll your own using the custom setting in the "LaTeX Options" compile pane, or create/modify .tex boilerplate files yourself.
- If you want to adjust how MMD generates figures in the output, then you will need to use XSLT to step into the process and exert control over the textual output of the image syntax.
- If you want to create you own output format from scratch, such as to say BBCode or Wikimedia, then you'll want to use XSLT.

22.6 Importing MMD Files

You can import any existing MultiMarkdown files into Scrivener using File ► Import ► MultiMarkdown File.... This will parse the file you select and break it up so that it is imported into the binder with its hierarchy intact, taking document names from the headers defined in the file, and using and non-header text that follows to fill in the main text for that section. Any meta-data at the top of the document will be placed into a file named "Meta-Data" at the top of the import. If this is done straight into the Draft, it would result in the meta-data being used in the final output. See MMD Meta-Data (section 22.7) for further details.

Illogical header depths, such as a level three header following a level one header, will be normalised to what is most logical. In that example it would become a level two document in the Scrivener outline.

22.7 MMD Meta-Data

MultiMarkdown supports a free-form meta-data system, with some meta-data keys being used intelligently by the system. You should read the MMD documentation about these fields. Beyond that you are free to add your own without detrimental effect. To set up project-level meta-data, which will be attached to every compile you produce, use the Meta-data pane in the compiler.

Additionally, you can add a document to the draft titled "Meta-Data". If this document is the very first document that Scrivener processes during compile (listed first in the Contents pane), it will be tacked on to whatever project compile meta-data has been defined. Owing to how MMD meta-data works, fields which duplicate will be ignored the second time. So if you wish to use both compiler meta-data and document meta-data together, you should keep in mind that if you have a "Title" in the compile settings, any "Title" in the document will be ignored.

When importing an MMD document via File ▶ Import ▶ MultiMarkdown File..., if Scrivener detects a meta-data block, it will create this file for you. If you wish to move these values to the compiler, or indeed if you wish to move any set of properly formatting meta-data values into the compiler, you can simply copy and paste into the compile pane meta-data table. Scrivener will read in the meta-data and convert it to the key-value system it uses in the compiler.

Since the only requirement for the meta-data document is that it be the very first document in the list of things to process during compile, you can use various features of the Contents pane to manipulate which meta-data is used during compile. A few examples include using Collections as your compile groups (where the meta-data file could be added to the top of the Collection), using the front matter feature to swap out meta-data sets, or using filters to selectively use the appropriate meta-data file for the content being delivered.

22.8 Updating MultiMarkdown

Scrivener comes with MultiMarkdown support built in, so you don't have to worry about installing anything extra. However, there may be times when MultiMarkdown updates in between Scrivener versions. In that case you may want to ensure that you have the latest version of MultiMarkdown installed. To ensure that Scrivener uses the most recent version of MultiMarkdown rather than the version built into Scrivener, you just have to install the new version of MultiMarkdown into either ~/Library/Application Support/MultiMarkdown. The installer for the support package will handle placement for you.

22.9 Further Information

Fletcher T. Penney, the creator of MultiMarkdown, is actively involved in supporting MultiMarkdown in Scrivener. He has generously contributed time and effort in writing tutorials and answering users' questions on the forums. See his web site and the Scrivener forums for more information. Thank you, Fletcher!

We also have an active MultiMarkdown section on our own software forum, where other authors have come together to share tips and tutorials. If you have any questions, it's one of the best places to go on the 'net.

Part IV

Final Phases

> Nothing stinks like a pile of unpublished writing.
>
> Sylvia Plath

Distilling your work into a final product is an essential task for any writing application. Scrivener approaches this problem from multiple fronts, giving you plenty of options for producing a manuscript, web pages, printouts, e-books, and quite a bit more. Most of these methods are functions of the compiler, a powerful export feature which will take the contents of your draft folder and produce a single document from the many pieces it is comprised of. Due to the level of control that can be exercised at this stage, learning how to export can be itself a part of the learning process, but one that will in time benefit your ability to write freely in whatever manner you choose.

Most word processors are designed for business use, and approach document creation with the philosophy of showing you a good approximation of the final product as you work. This integrated way of working means that you are at once presented with the tools for writing with the same level of priority as the tools for formatting and layout. This not only creates a more confusing and distracting interface, but can lead to writing in a constricted environment, where you must work in a format which you are required to deliver.

Scrivener's approach, as with many things, allows flexibility. You can work in the described manner to a degree, where the text as you write it is formatted to how it will be delivered (although it will never present a full working preview in all ways), but you can also choose to completely separate that aspect from your writing experience, and let the compiler handle the details of formatting for you, from how titles are named, all the way down to whether or not your paragraphs are first-line indented or spaced apart.

Fortunately, the distinction between these two ways of working can be made as fuzzy as you like. That is, you can gradually introduce more automation into the compiler as you learn how to use it, rather than having to make a big jump all at once to a new way of working. In the chapter on compiling, you will be introduced to several basic presets that ship with Scrivener. These are intended to be starting points, and will not address everyone's individual ways of working or every university's style standards. If you are accustomed to working in a word processor, you might want to start with the Original preset and work from there. If you are used to working in plain-text, or another workflow that does not regard formatting as part of writing, then you might be interested in trying one of the other more comprehensive presets, such as the novel standard manuscript, or 12pt Times presets.

The compiler can also be a tool for producing specialised reports, by selecting only portions of each item in the Draft to be included, such as just the title and its synopsis and meta-data. The enumerated outline preset demonstrates one such strategy, where an indented outline of all your draft titles will be exported as a file.

Many authors will be taking their finished drafts to a word processor, desktop publishing, or scriptwriting application for final post-production work. We will discuss several common applications on the market and how best to work with them in Scrivener.

Finally, we will also discuss more traditional methods of printing and exporting, as well as a few techniques you can use to add final polish to your manuscript. The topics that will be covered in this part are:

- Creating a Table of Contents (chapter 23): It is easy to create a static list in typical table of contents formatting, which can include dynamic page number references which will be calculated by your word processor upon opening the compiled file.
- Compiling the Draft (chapter 24): Discusses compile strategies, format compatibilities, and gives a detailed run-down of every possible compile option.
- Exporting (chapter 25): Transferring the contents of your project to your hard drive is a great way to back up your work outside of Scrivener entirely, or share bits of your project with other people. This chapter discusses the options available to you and how best to use them.
- Printing (chapter 26): There are two ways of printing in Scrivener. One operates as an extension of the compile feature itself, and is the preferred way to print proofs or final submission-ready manuscripts. The second method lets you print quick one-offs of any file in the binder, print index cards off of a corkboard, or information from an outliner.

Chapter 23

Creating a Table of Contents

With the exception of e-books, there is no support for inserting a dynamic table of contents into your draft, but there is a way to produce a static list of items that is cross-referenced to page numbers (for supporting formats) during compile, that uses Scrivener Links internally to make the list useful for the author as well.

This feature is mainly useful in conjunction with the PDF/Printing workflow and with the RTF-based formats when compiled and opened in a word processor that supports bookmarks and cross-references.

Using a table of contents of with page numbers and PDFs is generally more portable, since page numbers are baked into the file and do not rely on dynamic features that only some word processors support. If you intend to distribute the file to a number of people, and are unsure of what word processor everyone uses, PDF will provide the most consistent result between all platforms.

23.1 How to Create a ToC

Creating a table of contents is a simple process, but because it is a static list, you will probably want to save it for one of your final steps, as any changes in outline order or the addition or removal of sections will not be reflected in the list.

1. Select all of the items that you wish to have included in the ToC. It may be easiest to do this in Outliner view mode, so you can collapse and expand the various containers until the Outliner looks roughly like what you want the ToC to look like (speaking from a standpoint of content).

2. No matter which view you are working from, once all of the appropriate items are selected together, use the **Edit ► Copy Special ► Copy Documents as ToC** menu command.

3. Create, if necessary, an empty file in the Draft folder and paste the ToC list into this document. You may now format the result as you wish.

> **Pro Tip**: If you need to create a mini-ToC in the preface for each part of a book, as is common with technical guides, you can follow these instructions to produce a smaller scale list of sections. You will just want to select the relevant section instead of the entire Draft and paste the ToC copy into the preface area for each part.

The resulting list will be formatted with the name of the section on the left and a special tag on the right, with an amount of indenting applied to each line in accordance with its relative outline depth.

The tag, <$p>, will be replaced with a page number reference during compile. The title will also be examined and updated to match the visible name of the section as it will be compiled. For example, if the Formatting pane will replace the binder title with the word "Chapter" followed by a sequential number, then the Table of Contents link text will be changed to match it.

You also can create your own table of contents by hand. There is nothing special about the "Copy Documents as ToC" command that cannot be replicated manually. So if you do not like the default look, you can either adjust the formatting after pasting or generate your own ToC from scratch by using the <$p> token and linking it to the section you wish to reference with Scrivener Links. The title should be linked as well, if you wish it to acquire any prefix or suffix information. Read Linking Documents Together (section 9.5) for more information.

23.1.1 Things to Watch Out For

- When first opening the compiled document in Microsoft Word, you will need to generate the ToC numbers by running a test print preview once; they will appear as question marks until you have done so.

- This feature requires word processors capable of understanding bookmarks and cross-references. Page numbers may appear as question marks in the list if the software do not.

- If you are not using title generation in compile, and are instead relying on a formatted title within the draft text itself, you may find the "Copy Documents as ToC" feature less useful and might wish to create your own from scratch.

- If some items come out with no page number, check to make sure those items are included in the compile settings, and are set to output text. If the preview area in the Formatting pane type row that corresponds with that item type is empty, then there will be nothing to link to.

- The dot fill used between the title and the page number is an OS X text engine underlining feature which may not be visible in all word processors.

23.2 Contents in E-books

Both the ePub and Kindle compile export formats have an automatic built-in table of contents generator which should often be used instead of the methods described in this section. If you are publishing to an e-book platform and wish to set up a ToC, please read about the E-Book Options compile pane (section 24.13.3).

If you wish to produce a custom ToC for your e-book, a better method will be **Edit ► Copy Special ► Copy Documents as Scrivener Links**, since the menu command to copy as a ToC will insert page numbering tokens which are meaningless in an e-book context.

If you merely wish to move the automatically generated ToC from one location to another in the e-book, use the <$toc> placeholder tag in an otherwise empty document to position the ToC where you would like in the binder. Remember that many people get to this via a menu in their e-book reading software or devices, so where it appears is often not important. Placing the ToC at the end of the book is a common tactic as it keeps it out of the way.

See Also:

- Compiling the Draft (chapter 24)
- Linking Documents Together (section 9.5)

Chapter 24

Compiling the Draft

The main purpose of Scrivener is to provide a place that will help you write a long piece of text (whether novel, thesis or factual book), which may be structured as individual pieces in the binder but which can easily be output as one large file for working with "the rest of the world". Thus, you write, gather and organise the material for your manuscript in the Draft folder, and when you are ready to export or print the manuscript as a whole, you use **File ▶ Compile...** (⌥⌘–E) which takes the contents of the Draft folder, formats it as you specify, and outputs (or prints) it as a single document. Using the various settings available, you can export or print your texts however you like—even regardless of how the files are formatted in Scrivener itself.

Compile settings are an intrinsic part of your project; they're just as much a part of it as any folder or file in your binder. They will be saved with the project and travel with it if you move it to another machine or share it with a collaborator. The available compile presets (which will be discussed shortly) are global to your machine, and can be applied to any of your projects, but once they are applied those settings are in your project. You can also save custom setups as your own presets to be used in this same fashion, or export them as portable files which can be used to share settings with colleagues, or update a second computer.

At its most basic usage (Figure 24.1), you can select from a number of handy presets using the **Format As:** drop-down menu on the top, and then supply a target file format using the **Compile For:** drop-down menu at the bottom. In between, you'll find common options pertaining to the file format you have chosen. For example, with the e-book formats you can quickly set up a cover image. Some of the presets use more detailed options that cannot be seen initially. If you require more detailed control of the exported file, you can click the **All Options** tab to reveal the full Compile interface. From here it is possible to control practically every element of your draft as it is processed into a single file.

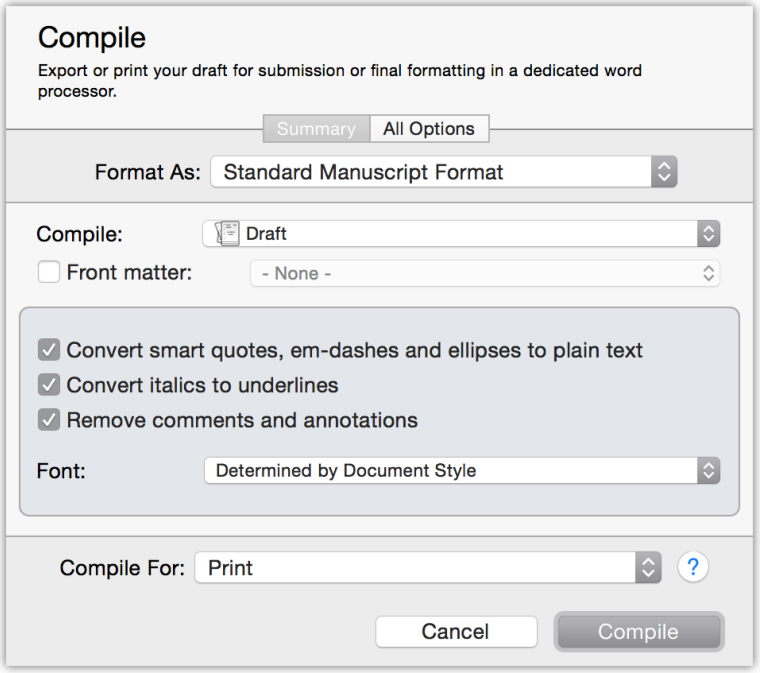

Figure 24.1: Compile interface in summary mode.

24.1 Compile Summary

The available settings are determined by the export format (section 24.3), chosen in the "Compile For" drop-down menu. Paper and margin settings, for instance, would serve no purpose when compiling to plain-text, so it will be removed when using this export format.

The settings here will correlate to a particular setting (or set of like settings) in the "All Options" tab, and changing one will affect the other. Summary view is just that, a quick list of common options readily available, but not representative of the entire compile setup, which may perform functions that are not displayed in summary view.

The following lists all of the possible options available, sorted by export format, and cross-referenced to the full option pane whence it came.

Common Options These are included for all compile export formats.

- Compile (section 24.7): selects the source material for the compile. Ordinarily this is the entire Draft, but you can also choose from portions of the Draft, Collections, or even based on your current Binder selection.
- Front Matter (subsection 24.7.3): lets you pick from a folder of alternate front matter material that you have set up in the binder. Use this option to quickly

switch between different front matter sets depending on how you are compiling. Check the box to enable the feature and then use the drop-down menu to select the folder which contains the relevant front matter for your document.

- ❧ Remove comments and annotations (section 24.21): disables the output of all commentary, either inline or in the inspector.
- ❧ Convert smart quotes, em-dashes and ellipses to plain text (section 24.15): Enables or disables all three of these common text transformations. When viewing all options, these can be set independently. While this option is used in the e-book formats, it will not be displayed in summary view. Switch to "All Options" and use the "Transformations" pane to make changes to the plain-text conversion options.

Rich Text, Web Pages & Printing This includes Print, PDF, RTF, RTFD, DOC, DOCX, ODT, HTML, and WebArchive. * Font (section 24.26): set this to "Override all fonts with face" and choose a font to completely override every font setting. This doesn't change the font settings themselves, but rather a special override which adjusts body text, titles, extraneous material (like Notes and Meta-Data), headers and footers, and footnotes. Switching this option back to "User current compile format font settings" will disable the override, and the original font selections will be used. * Convert italics to underlines (section 24.15): if the submission process requires underscoring to be used instead of italics, this feature will let you write in italics but produce a properly underscored manuscript. To convert underlines to italics, you'll need to view all options.

Plain-text (TXT) ❧ Convert to plain text (subsection 24.15.1): depending on the choice made, actual spaces and carriage returns will be inserted into the document to emulate indents and paragraph spacing (line spacing will be ignored), as well as other effects.

Final Draft (FDX) ❧ First document is title page (section 24.14): set this if the first document in your draft is a title page, so that it can be marked as such for Final Draft.

- ❧ Include footnotes and comments as script notes: all notation styles will be exported as Final Draft style script notes.

E-books This includes EPUB, and MOBI.

- ❧ Title (subsection 24.24.2): the title of the work. This is what will show up in book lists when browsing through books in an e-reader.

- Authors: this will show up when browsing by author in an e-reader.
- Cover image (section 24.10): select from any graphic in the Binder to use as the e-book cover. This is typically placed on the first page of the book in the e-reader, and some software will also use it when displaying the book in a list.
- Generate HTML table of contents (subsection 24.13.3): causes a copy of the table of contents to be formatted and printed in the e-book material. A "meta-data" version of the table of contents will always be encoded into the e-book for use by readers that support quick navigation.

MultiMarkdown This includes MMD, MMD-HTML, MMD-LaTeX, and MMD-RTF.

- Convert comments and annotations to HTML-styled text (section 24.21): causes Scrivener to insert spans and divs around inline and inspector notation. This will cause text to appear coloured as per the original note, in most cases. Turning this off will cause comments to appear as ordinary text wrapped in square brackets. This option is only available when notes are not being completely removed.

24.2 Built-in Presets

In some cases, you may not need to look any further than the supplied presets that have been bundled with Scrivener. Some of these have been designed to conform to common industry standards in terms of manuscript submission and working with agents and editors; others have been designed as useful working tools, such as the ability to export an indented outline, or the distribution of proofreading copies.

If you wish to customise these presets, you will want to read the section on Expanded Compile Interface, directly following, and any of the Option Panes which are relevant to you. Once you have changed a preset and saved or compiled, the selection menu will revert to "Custom". Your custom settings will always be saved in the project, but if you change presets you will lose your custom settings. If you wish to save your settings for future use, read the following section (section 24.6) for instructions on how to do so.

In addition to these global presets, many of the project templates that ship with Scrivener come with useful compile settings as well, saved into the projects that are created from those templates. You can always recall the original compile settings used by a template, using the Format As drop-down menu, and selecting the template's special preset from the list.

The available built-in presets are:

- *Original*: This preset is the easiest way to use the compiler. If you feel overwhelmed, it may be more efficient to just compile to a basic word processor format like RTF, using the "Original" preset, and finishing off the formatting in your preferred word processor.

  This is the standard compiler setting, and in a sense it isn't really a preset, just a complete lack of one. It will preserve your draft as you wrote it, without adding any titles, formatting adjustments, and so on. If you have very carefully designed your book from the start, this might be the best option, or at least the best starting point.

  This option will appear at the top of the menu when starting from a blank project, but will be moved down the list, next to the special "Custom" entry (which holds your current settings), when the project was created off of a template. In that case, the template's original formatting will be stored in an inviolate preset at the top of the menu.

- *E-book*: A suitable, simple format for e-books. ePub and Kindle files generally require basic and flexible designs in order to be displayed on many devices, from cellular phones to tablet computers to dedicated black & white e-ink displays.

  - This format assumes all containers to be chapters, and all text files to be scenes or sections.
  - Use the "With Parts" counterpart to this preset if you have organised your chapters folders into part or book folders.

- *Enumerated Outline*: If you need a basic outline of topics alone in an indented list, this preset is a good starting point. Unlike most compile presets, it will not export any text, only the titles of the documents. Hierarchal numbering (1, 1.1, 1.1.1) will be used for all levels of the indented outline (up to six levels of indentation, though you can add further levels if you require). It has also been configured to accommodate synopses, which you can optionally enabled in the Formatting pane (section 24.11). This is a useful starting point if you want a simple data sheet print out of all the pieces in your Draft folder.

- *Non-Fiction Manuscript Format*: Treats all folders and top-level file groups as chapters. Top level files will be titled, and any files below the top level will be untitled, among other tweaks to bring it more in line with common style practices in academia.

- *Paperback Novel*: A format designed to produce a typographically pleasing result which is fine-tuned for PDF and Print output. This will produce a PDF that is suitable for submission to many POD services, such as CreateSpace.

  Use the "With Parts" counterpart to this preset if you have organised your chapters folders into part or book folders.

- *Plain Text Screenplay (Celtx, Movie Magic)*: For integration with software that can import plain-text formatted screenplays[1].

- *Proof Copy*: A useful preset for internal proofing prints. It will reformat your script to double-spacing so you can easily take notes, and print a disclaimer after each chapter title as well as in the header, making it easy to send out "Not for distribution" copies to your proofing team. This preset has been set up to treat folders as chapters and everything else as sections. If your book is organised differently, you will want to change how things are arranged in the Formatting pane.

- *Script or Screenplay*: Represents common settings for integration with Final Draft version 8 or greater, using the FDX file format.

- *Standard Manuscript Format*: Formats your book using standard Courier 12pt type and a number of common conventions such as scene separators, double-spacing, underlined emphasis (instead of italic), page numbers, and so forth. Note to get the full benefit of this preset, you will want to use one of the rich text formats, such as RTF. This preset has been set up to treat folders and top-level files (though not file groups) as chapters, with everything else being treated as sections. You will want to adjust this in the Formatting pane, if your book structure differs.

- *Synopses and Titles*: Similar to "Enumerated Outline", without the hierarchal numbering prefixes, and synopses have been added to all file types by default.

- *Synopsis Outline*: Presents an indented, easy to read outline that includes titles and synopses for all binder levels. Files at levels greater than the top level will not print their titles by default.

In addition to these, a number of optional presets have been included, but disabled so as to not clutter up the main format preset menu. If any of these following interest you, you can find and enable them in the Manage Compile Format Presets... menu choice of the Format As drop-down.

[1] Not to be confused with Fountain, which is a newer format with a rigid syntax, this format is more along the lines of how you would type in a screenplay using a typewriter, with tabs and spaces.

- *NaNoWriMo (Obfuscated)*: this simple preset treats your document nearly like Original does, though it will generate titles for folders and file groups. It uses plain-text, which is what the NaNoWriMo project needs for verification of your word count. It also comes with an extended list of letter-to-letter substitutions which will completely obfuscate the wording of your draft, making it nearly impossible to read, without impacting the word count.
- *Non-Fiction with Sub-Heads*: Similar to "Non-Fiction Manuscript Format", except with more extensive sub-titling treatment down to four levels of outline depth for all file types.
- *Non-Fiction with Sub-Heads (Hierarchal)*: Same as above, but uses a typical heading prefix numbering scheme as found in many technical works.
- *Times 12pt with Bolder Folder Titles*: This is a legacy preset which was used to offer an alternative to the standard manuscript format for those that required Times New Roman manuscript. This can now be achieved by using the quick font override option with the standard manuscript preset.

24.3 Available Compile Formats

The compile formats table (Table 24.1) shows all the formats supported by Scrivener, with commentary on their usage.

For most word processor-based workflows, including Microsoft Word, RTF is the best option. Nearly every word processor provides solid RTF import and export capabilities, and some even use RTF as their native file format. The one major exception to this is Pages, which has limited RTF support. You should use .doc or .docx with pages, and for the best results the improved converters should be enabled (section B.10).

> **RTF Files Missing Some Features?** With the default OS X configuration, when you double-click on RTF files in the Finder, they will be opened in TextEdit, which doesn't support all RTF features. To open an RTF file in your desired word processor, you should open the word processor first and use File ▸ Open.... You can use the "Get Info..." palette in Finder to change the software association with RTF either for one or all files.

24.3.1 Using RTF

The provided table (Table 24.3) shows which file format constitutes the best option when exporting for use with several popular applications, and which features are supported by those programs. If you are using the improved converters, then select the format

Table 24.1: Compile Formats Table

Format	Extension	Description
		General Purpose Formats
Print	N/A	Used to immediately print the compiled draft
PDF	.pdf	The Portable Document Format is a (largely) uneditable format that can be opened on nearly any platform and device with minimal to no loss of display quality.
Rich Text	.rtf	General purpose rich text format supporting multiple fonts, images, tables, bullet points, footnotes and comments. The standard internal format for Scrivener, and the base underlying format used for most conversions in this table.
Rich Text with Attachments	.rtfd	Apple's proprietary extended RTF format. Useful mainly for exporting to other Apple Cocoa applications such as TextEdit, especially if image support is needed. Incompatible with most word processors and all other computing platforms.
Microsoft Word 97–2004	.doc	Binary Microsoft Word document format. Use this if the target application cannot read RTF or DOCX files.
Microsoft Word	.docx	Microsoft's modern XML based Word document format; preferred for usage with Word and compliant word processors.
OpenOffice	.odt	The OpenDocument Format is a free full-featured XML document format, supported by many word processors, including OpenOffice and Google Docs.
Plain Text	.txt	UTF-8 (Unicode) plain-text file. Plain text contains no formatting but can be opened almost anywhere, on all platforms and devices.
		Scriptwriting Formats
Final Draft	.fdx	For transferring your script to Final Draft, this is the best option unless you have an older version of Final Draft than version 8. Maintains synopses (as scene summaries), scene titles and custom script element formatting.
Final Draft 5–7 Converter	.fcf	Use the FCF converter when working with an older copy of Final Draft. Supports only basic screenplay formatting.
Fountain	.fountain	A plain-text screenplay markup format ideal for working on the go.

Table 24.2: Compile Formats Table Cont'd

Format	Extension	Description
		E-book & Web Support
ePub eBook	.epub	Generate feature-rich e-books for use in portable reading devices that support the ePub format, such as the Sony Reader, Nook, Kobo or many iOS based readers (ePub files can be dragged into the iTunes Library to import them into iBooks).
Kindle eBook	.mobi	Generate feature-rich eBooks for use in portable reading devices that support the .mobi format, such as the Amazon Kindle. Requires Amazon's KindleGen, which is only available for Intel based Macs.
iBooks Author Chapters	.docx	The .iba format cannot be directly supported, so this method will export all sections of the book (going by page breaks) to individual .docx files, which can be easily imported into iBooks Author.
Web Page	.html	Creates a single HTML file suitable for web-publishing.
Web Archive	.webarchive	Much like HTML, but conveniently packages exported image files into a single bundle, using Apple's webarchive format, which can be opened by Safari (including the Windows version) and various Mac OS X applications. Not compatible with most other browsers.
		MultiMarkdown Post-processing
MultiMarkdown	.md	Export a plain-text MultiMarkdown file, useful for archiving, or further custom post-processing.
MultiMarkdown -> LaTeX	.tex	Exports a LaTeX format file with full MultiMarkdown parsing. Note that if you are intending to export a LaTeX file that has been handwritten in Scrivener (without MMD), you should use the plain-text exporter, above.
MultiMarkdown -> RTF	.rtf	Export an HTML-derived RTF file with partial MultiMarkdown support. Note that the RTF files generated in this fashion will be limited in features and formatting support in the same way as HTML, and is not equivalent to Scrivener's standard RTF exporter in the word processor compatibility tables.
MultiMarkdown -> Web Page	.html	Generates a W3C compliant HTML5 file, suitable for web-publishing or further XML post-processing.
MultiMarkdown -> Flat XML	.fodt	This ODF compatible format is suitable for bringing your MMD project into the word processor world. At the time of this writing, LibreOffice is the main OpenOffice fork that can read .fodt files without modification. Nisus Writer Pro can read .fodt files, but only if the file extension has been change to .odt.
MultiMarkdown -> PDF	.pdf	Using the LaTeX typesetting engine, produces a simple ready-to-print PDF file. This option will not appear unless pdflatex has been installed on the system.

appropriate to the word processor (although RTF will most likely be just as good, and compile faster, so try that first). If the word processor you favour is not available as a specific format, use the table to select the best format. The one exception is iWork Pages. You should use DOCX with Pages unless you are using the basic converters.

Table 24.3: Supported Features by Word Processor for RTF

Application	Best Format	Comments	Footnotes	Lists	Images	Head/Foot	ToC
Microsoft Word	RTF	X	X	X	X	X	X
Nisus Writer Pro	RTF	X	X	X	X	X	X
OpenOffice.org	RTF		X		X	X	X
RedleX Mellel	RTF	X	X	X	X	X	X
Papyrus	RTF		X			X	X
Apple Pages	DOCX	X	X	X	X	X	X
TextEdit	RTFD			X	X		

Explanation of features and how compatibility is determined:

- **Comments**: RTF comments are an optional way to export Scrivener's comments and annotations. Compatible word processors will display these comments in their own fashion, usually in the margins. If annotations are exported as inline comments then they will be supported by all editors, as they just become ordinary text at that point. This is the recommended setting if your word processor does not support RTF comments, but you need to export annotations.

- **Footnotes**: For compatibility, so long as a word processor supports footnotes and/or endnotes, it is considered compatible. Not all word processors support some of the more advanced footnote appearance features, such as custom numbering styles and restarting numbering on each page.

- **Lists**: Compatibility is considered to be simple bullet and enumeration list transfer. Specific bullet styles and custom enumeration styles may not be equally supported by all programs.

- **Tables**: Only basic table support is required to be considered compatible. Custom styling and other advanced features are not taken into consideration. Most word processors do *not* support nested RTF tables, but do support cell background colour and border options.

- **Images**: Inclusion of inline images is required for compatibility.

- **Headers/Footers**: Page headers and footers; custom first-page settings are not considered for compatibility.
- **ToC**: Support for Scrivener's Table of Contents feature, including RTF Bookmarks. While technically two different features, word processors that support one will usually support the other, so they have been combined.
- **Note on RTFD Support**: Since the two applications for which RTFD format is recommended do not support these RTF features anyway, it often better to use RTFD with them, as they will benefit from a few extra features that would otherwise be dropped. RTFD will "flatten" some of the features it does not support, such as annotations and footnotes.

Not included in the table are basic formatting features. Most word processors support these with only a few exceptions, as listed below (if the word processor is listed next to the formatting feature, it does not support it):

- Text Highlights: OpenOffice.org
- Background colour (page): OpenOffice.org[2]; Nisus Writer Pro; Mellel; Papyrus
- Custom underline styles (dotted, dashed, etc): Nisus Writer Pro; Papyrus
- Kerning: Mellel

24.3.2 Improved Import and Export Engine

Included in version 2.3 and greater, the Aspose document conversion engine is used for importing and exporting Microsoft Word (.doc and .docx) and OpenOffice (.odt) formats. The Aspose engine uses Java, which comes provided for you with the Scrivener download, and enabled by default. This module will provide the ability to compile .doc/x or .odt files at the same level quality as the RTF format.

> **Should You Use the Improved Converters?** If RTF has worked well for you in the past, there is no need to change your habits. You will not gain anything by doing so—Scrivener in fact starts with RTF, and uses the Aspose engine to convert that RTF to the final format for you. Thus for most usages, RTF is still the preferred format; it is readily supported by most word processors and fast.

[2] Including derivatives, such as NeoOffice and LibreOffice.

> However, some applications (particularly Apple Pages and mobile office suites) cannot read RTF, or do a poor job of it. In these cases you will want to use a Word format instead, and the Aspose converters will provide the best possible option for doing so.

If you would prefer to not use the improved converters, you will need to disable them in the application Import/Export preference tab (section B.10), not the compiler. This switches the feature off globally. They will no longer be used for import or export in any way. Instead, the basic Mac OS X converters will be used, which have a lower level of quality, but do not use Java and are faster as a result.

24.3.3 Portable Document Format (PDF)

PDF files offer a great way to archive and distribute copies of your work. They look great almost everywhere you open them, can be opened on nearly anything from smartphones to workstations, and are—while not technically, effectively—static. That is, they cannot easily be edited, making them good for archival and proof distribution. Many PDF readers also come equipped with annotation tools.

There are two important modes for generating PDFs to be aware of, in the Print Settings compile option pane (section 24.8). If the custom converters (section B.10) have been enabled, this pane will appear and provide a choice between "Publishing" and "Proofing". These are general terms, so you should read the full implications of choosing either mode when making your selection. The default is "Publishing", which follows the model of prior versions of Scrivener. This pane is also where URL based hyperlinks can be configured.

PDFs created using the Publishing layout engine will by default include a software outline, commonly used in PDF viewers to display a navigable table of contents in a sidebar or menu.

Exported PDF files (both Publishing and Proofing) can also contain clickable internal links, taken from any Scrivener Links in the text. Going back to the notion of a printable table of contents, if you followed the directions put forth there, you will have a list of binder item names linked to the original names in the binder. In the PDF, these names will be hyperlinked to the correct spot within the PDF itself. With the Publishing layout engine, you can modify how this is done, in the compile options pane, PDF Settings (section 24.17). Since a table of contents created in this fashion is merely a formatted list of Scrivener Links, it follows that any links you create within the main text itself will also be linked.

24.3.4 Exporting Scripts

The following table shows which file format constitutes the best option when exporting for use with several popular scriptwriting applications.

Table 24.4: Script Format Recommendations

Application	Best Format	Notes
Final Draft	FDX	Supports comments and footnotes (as script notes), synopses (which become scene summaries), titles, dual dialogue (dialogue marked using "Preserve Style" in Scrivener becomes dual dialogue in Final Draft), revision marks, custom element formats.
Final Draft 5–7	FCF	FDX is always better, so only use this if you don't have access to Final Draft. FCF is essentially a plain text format and so any formatting such as bold or italics will be lost. Also note that it only supports the basic script elements—Scene Heading, Action, Character, Dialogue, Parenthetical and Transition. See below for important tips which will help avoid strange characters appearing in the export.
Fountain Markup	fountain	Fountain is a Markdown inspired plain-text format, and is thus suitable for writing screenplays on a wide variety of devices. Scrivener will convert its built-in scriptwriting format to Fountain's markup automatically.
Movie Magic Screenwriter	TXT	See below for important tips which will help avoid strange characters appearing in the export, and ensure proper element conversion.
CeltX	TXT	As for Movie Magic Screenwriter.
Montage	RTF or TXT	Montage will do a decent job of importing script files saved in either the RTF or TXT formats (if you use TXT, follow the same rules as for Movie Magic Screenwriter and CeltX).

The guidelines below should be followed when exporting to the FCF or TXT formats for use with scriptwriting programs:

To avoid strange characters appearing in the export, the following preferences should be enabled in the Text Options compile pane:

- Straighten smart quotes

- Convert em-dashes to double-hyphens
- Convert ellipses to triple periods

In the Separators pane:

- Set all separator types to "Single return".

When using the TXT format, along with the above settings, "Convert indents and paragraph spacing to plain text" should be enabled when exporting to scriptwriting programs such as Movie Magic Screenwriter and CeltX. This ensures that they will recognise the elements correctly, as these programs read the whitespace and use it to convert the text to the relevant script types.

Read Working with Final Draft (section 20.5) for more detailed information on both export and import processes with Final Draft 8 and higher.

24.3.5 iBooks Author

The iBooks Author format cannot be directly saved to. As is the case with Pages, Apple has not released any support for third-party applications. Fortunately, it is fairly easy to import material into the program. Given how it is designed, it works better if each major section of the book (such as a chapter) is organised into individual files, rather than importing one large document; its capacity to split up an existing imported document is rather limited. If you intend to publish your book using iBooks Author, you should use the "iBooks Author Chapters" compile format. This will cut your book into separate .docx files wherever a page break appears in the text. You can then just drag the entire folder of compiled files into iBooks Author and continue your work in that program.

The key to this is to make sure that page breaks happen where you expect them to. This is typically handled in the Separators compile option pane (section 24.9), with the "Page Break Before" inspector checkbox being used in those cases where Separators cannot address outliers like a preface, copyright page, or table of contents. When using the iBooks Author format, the Separator's pane will list "Chapter Break" as an option, instead of "Page Break". This setting marks where one file will end and the next will begin.

24.4 Compiling and Saving Settings

Once you have everything set up the way you like, the **Compile** button will automatically save all of your settings into the project, dismiss the window, and compile the project in accordance with your specifications. The important thing to know here is that each

project's compile settings are saved as a part of its persistent settings. When you change settings and compile, they are saved back into the project and will stay that way until you change them. There is nothing special you have to do beyond this to save your export settings. In a moment we will discuss presets, which are a way of saving *sets* of settings for future use, so that they can be applied to projects. Thus, you only ever need use presets for the purpose of backing up or transferring settings between projects, since settings are automatically saved otherwise.

You can cancel any changes you've made by clicking the **Cancel** button at the bottom of the compile window. This will exit the dialogue without compiling, and in addition, it will reset any changes you've made to the compile interface since you loaded it.

At any point, you can hold down the Option key in the compile sheet. This will switch the **Cancel** and **Compile** buttons to **Reset** and **Save**, respectively. Reset will revert all changes you have made back to the last saved version (or preset default if you have never used compile for your project before). The **Save** button will dismiss the Compile sheet, but save your settings before doing so. This is useful when you want to make a change to your export settings, but do not want to actually compile the document yet.

24.5 Managing Custom Presets

Since compile settings are saved into the project whenever you compile, or hold down the Option key and click the revealed **Save** button. There is no need to repeatedly save to a preset in most cases.

So when should you save compile settings? You might wish to save them for use in other projects, or simply to temporarily use a special-purpose compile preset (like *Enumerated Outline*) without losing all of the work you have put into your settings.

There are two tabs at the top of the preset dialogue, "Global Presets" and "Project Presets". As you might guess, saving global presets will add them to the main **Format As:** drop-down menu at the top of the compile sheet, meaning from that point on, you can apply them to any other project you create. Project presets will be added to the same menu, however they will only be visible from the current project you are working on. So this makes a great choice when you merely need to switch between multiple output profiles for a specific project—for example if you are publishing both an e-book and a PDF—but know that the specifics of this preset will never be useful to other projects. You can keep the main list tidy by using the Project Presets tab, instead.

> **Transferring Presets Between Tabs**: In the past, you may have used this feature to save project specific presets, but did not have the option to use the separate tab. If you would like to move these specific presets over to the project tab, you will of course first

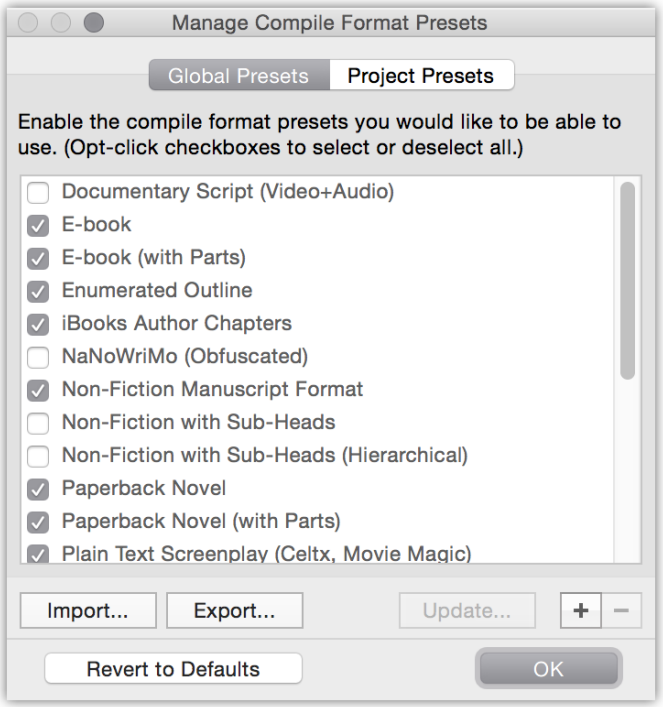

Figure 24.2: Manage Compile Format Presets

> need to load the project that you wish to install the presets into, and then use standard Copy and Paste actions in the Manage Compile Format Presets window, between tabs.
>
> Likewise, if you have created a project preset and realised later on that it should be available globally, you can use Copy and Paste to "promote" it to the Global Presets tab list.

To access preset management features, select the last option in the "Format As" drop-down menu, **Manage Compile Formats...**. A pop-up window (Figure 24.2) will appear, presenting a list of all available presets, as well as buttons along the bottom for managing them. Built-in presets are marked in grey text, and some of the functions for managing them will be disabled. You cannot, for instance, permanently delete a built-in preset.

The checkbox beside each preset governs whether or not it will appear in the main "Format As" menu. This way you can store as many presets as you want, even if you aren't using them regularly, without cluttering up the preset menu. You can toggle all on or off by holding down the Option key while clicking on any checkbox.

Import... brings up a file browser. Locate a saved compile preset on your drive to install it into Scrivener.

Export... The selected preset in the left list will be saved as a file on your disk which you can then use to carry to another machine, send to a colleague, or back up for safe keeping.

Update... Will transfer the *current* compile settings for your project into the selected preset. This is useful when you've made incremental changes to one of your presets and wish to update it for future use.

Built-in items cannot be updated. To create a variation of a built-in, first apply the settings using the main Format As drop-down, make your desired changes, and then use the management window to create a new preset with the + button.

Plus & Minus Buttons Clicking the + button presents a dialogue box in which you can title the preset. This is how it will appear in the preset menu, so it often works best to choose a meaningful name that you will remember in the future. You can also use the delete key.

Clicking the - button will delete the selected preset. Built-in items cannot be deleted (to disable them, use the checkbox).

Revert to Defaults Clicking this button will reset the checkboxes to the factory default, it will not make any changes to the list itself.

24.6 Expanded Compile Interface

The full compile interface (accessed by clicking the All Options tab) is where fine-grained tweaks can be made to the compile process.

The full compile window is arranged into two main sections. On the left is a list of options pertaining to the currently selected export format, at the bottom of the window. Clicking on an option title in this list will reveal an associated configuration screen on the right side.

24.6.1 The Option Panes

Option panes are along the left side of the window, in a sidebar style list, when the "All Options" tab is clicked. Each export format has a variety of features available for customisation. Some of the tabs will be available to all formats, but some are only available to certain types, and others only appear for single specialised formats. Since there is a good deal of duplication, this section will go over each of the available tabs only once, roughly in the order which they would appear in the sidebar list. It will be noted in each section which formats the tab is relevant to.

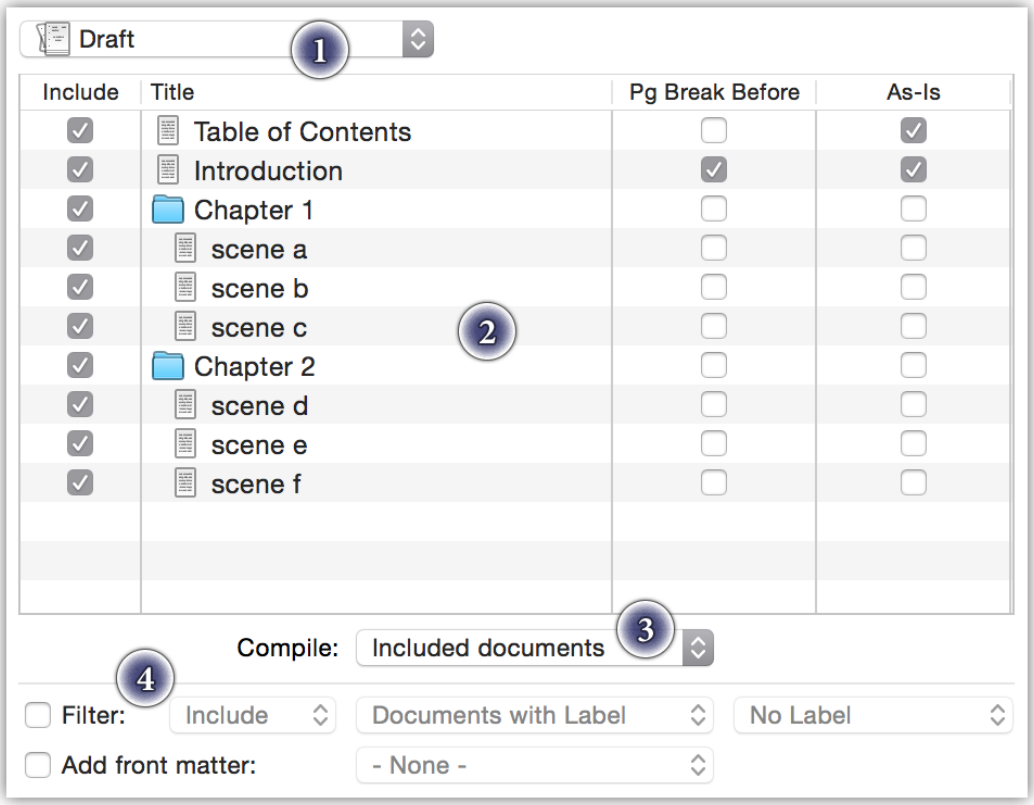

Figure 24.3: Compile—Contents Pane

24.7 Contents

This option pane is a core feature available to all export formats.

The Contents pane is where you define which parts of the Draft Scrivener will use to create your exported file. To accomplish this, it provides several tools of varying scope. The method with the most immediate impact is the drop-down menu at the very top of the interface, marked as 1 (Figure 24.3), which sets the "Compile Group". Ordinarily this will have the Draft selected (which might be called something else if you have renamed it, or started with a template).

In cases where you wish to proof only a portion of the book, or are working in a compound project that includes several editions or articles located in the Draft, you can use this drop-down to select only a portion of the draft folder. Whatever container you select will include everything from that point downward, or in outliner terminology, its descendants. You could, for example, choose the folder called "Chapter 1". All of the scene files included in the folder will be used in the Compile, optionally including the chapter folder itself, but nothing outside of that folder will be used.

Depending upon the type of selection made with the content drop-down menu, secondary options may appear alongside it.

When selecting a component of the Draft A secondary drop-down menu, "Compile Group Options" provides for two separate options on how the grouping should be handled.

- *Treat compile group as entire draft*: ordinarily when a portion of the Draft is selected for compile, the formatting treatment and counter numbering will be displayed as though the rest of the manuscript existed; chapter 13 will remain 13. When this option is enabled, the smaller section of the Draft will be treated as though it were the entire manuscript. All counters (either in compile settings or in the draft itself) will start at 1, and the immediate children of the items of the selected compile group container will be treated as level 1 items in terms of formatting styles.
- *Include selected container in list*: the selected container is included in the compile group by default. If you would prefer to only see the items beneath it and downward, then turn this option off. When the above option has been set as well, this can cause level based formatting rules to shift up by one, as the items within it are no longer level 2, but level 1. When the above option is off, all levels are computed as though the rest of the draft folder were included, and so this option will make no difference in formatting.

When using the current selection This option uses the active selection in the main project window (which could be cards in a Corkboard, or items in the Binder or a Collection). As with Collection based compile groups (below), this option will produce a flat list for quick compile. If you wish to use your selection as a filter (subsection 24.7.2) against the main list, and thus preserve hierarchy.

The secondary option, "Include subdocuments" will set whether or not your selection should automatically include anything beneath the selected items, too. This is especially useful if you wish to compile two folders with many sectional files. You can just select the two containers, click this option on, and be done with it.

When selecting a from Collection In addition to folder selection, you can also select a Collection to be used the content source. When Collections are in use, there are two important things to consider:

1. The rule of thumb in which only items within the Draft can be compiled will be temporarily dismissed. Items placed into a collection from outside of the

Draft *will* be included in the compile. However, only text and folder items will be used.

2. Since collections have no inner hierarchy, the compile will always result in a flat list of documents. Book parts will be on the same level as sub-sections, to put it into practical terms. Since the final result will always be a flattened file anyway, the main thing to note here is that only those formatting stylesheets relating to Level 1 will apply.

Collections have no secondary options available.

24.7.1 Content Item List

The item list is the large table in the middle, displaying the contents of the current compile group. In the case of the illustrated example, marked as 2 (Figure 24.3), the Draft has been selected, and so the entire Draft contents are revealed in the list below it. The items in this list will be indented in the same fashion that they are indented within the binder, representing their hierarchy. If you wish to use Collections or a selection to group by, but still retain hierarchy, read the next section on filtering.

There are four columns in this list. It is usually a good idea to initially scan these columns to make sure everything will act in the manner you expect it to. On any of the checkbox columns, you can Option-click the header to toggle checkmarks for all visible items in that column.

Include Linked to the Inspector option, "Include in Compile", if unchecked the item will not be used in the final product unless the compiler has specifically be instructed to use this checkbox in an unorthodox fashion (below). This is generally used to create static exceptions for items which will rarely or never be a part of the compiled product. For quickly filtering or selecting the scope of a compile to for example print out one or two chapters, you should use the compile group selector or filters.

Title The visible name of the item in the Binder. This might be altered in subsequent compile options, or updated with a counter if it includes a token in the name.

Pg Break Before Linked to the Inspector option, "Page Break Before".

Inserts a page break control before the checked document. A common usage for this is to set special pages like title pages and table of contents. For items which create a repeating pattern where a page break would be needed, you should use Separators (section 24.9) instead of individually marking checkboxes for each item; and indeed most default settings include an automatic page break wherever a folder

follows a file (often used to denote a new chapter). With some compile formats, page breaks can be important, particularly the e-book formats, which use section breaks to determine the automatically generated table of contents.

As-Is Linked to the Inspector option, "Compile As-Is".

All formatting set in the formatting pane (section 24.11) will be ignored for the checked document. This also means no extra material will be added, including titles or synopsis. Text will always be included, even if it would otherwise be excluded.

In the lower-right corner (marked as 3) is an override which can reverse the settings of the "Include in Compile" checkmark. By default, the behaviour follows the logic of the document settings. You can however inverse the logic and only export those items marked to *not* be ordinarly exported—or just ignore the flags altogether and export everything.

24.7.2 Filtering

Filtering is an advanced feature, near marking 4 (Figure 24.3) which uses the current compile group and applies additional criteria to filter the list down further. This works in a similar manner to using Project Search: only those items which match the criteria will be included in the compile group list.

As with the compile group drop-down menu, the changes you make here will impact what you can see in the list. If when you first load up compile and are puzzled by an empty list, check the Filter options and make sure nothing has been left set there from a prior session.

Filters can be defined in one of two ways, set with the first drop-down menu:

- *Include*: Everything matching the rest of the filter will be included in the compile, non-matching items will be removed. This is the default setting.
- *Exclude*: Everything matching will be *removed* from the compile.

The second drop-down specifies which attribute you wish to filter by. There are four options:

- *Documents with label*: documents matching the Label specified in the third drop-down menu will be matched and handled according to the logic in the first drop-down menu. This is the default setting.
- *Documents with status*: as above, only using the Status meta-data field instead of the Label field.

- *Documents in collection*: will provide a list of available Collections for you in the third drop-down. Items found in that collection will be included or excluded depending on the setting in the first drop-down.

- *Current selection*: as with the content selection item, this will use the current selection that has been made in the sidebar, prior to opening the compile interface. However, instead of producing a flat-list, the selection will be used as a filter, based on the above logic of include or exclude, and thus preserving the original structure. Since this method has no optional behaviour, the third selection drop-down will be removed.

Applied Example: Meta-data Filtering

In the provided example (Figure 24.4), the Contents pane has been simplified to only show relevant interface elements. Additionally, the option to show label colour (here, renamed to "PoV") has been enabled in View ► Use Label Color In ► Icons.

Example 1 has the Filter checkbox disabled. The list is thus a pure depiction of whatever container has been selected for compilation, in this case the top-level Draft. We have labelled scenes a, d, and e with the "Joseph" character colour, and scenes b, c, and f with another character. The other items have no colour label.

Example 2 shows the Filter option enabled, set to *include* all *documents with PoV (label)* of *Joseph*. The resulting list has been filtered to only show the three items which have had the "Joseph" label set. What you cannot see here is that it is only selecting items within the current compile group. If other documents in the project are labelled "Joseph", but outside of the current compile group, they will not be included. Filtering only adjusts the current contents of the compile group.

Example 3 is more subtle. In this case we have left the second two options alone, but changed the first option to *exclude* all *documents with PoV* of *Joseph*. Note that not only have the scenes from the PoV of the other character been selected, but all of the items with no label set at all.

The manner in which the status meta-data filter works is identical to the label meta-data, so we will move on to the next example.

Notes on Collection Filtering

The basic premise behind Collection filtering is the same as when using meta-data fields. Items which match the criteria will be included, or if *exclude* has been set, everything except those matching items will be included. Also as we demonstrated with meta-data, the candidate items available for filtering must originally appear in the unfiltered compile group list. Even though Collections can contain items from anywhere in the project,

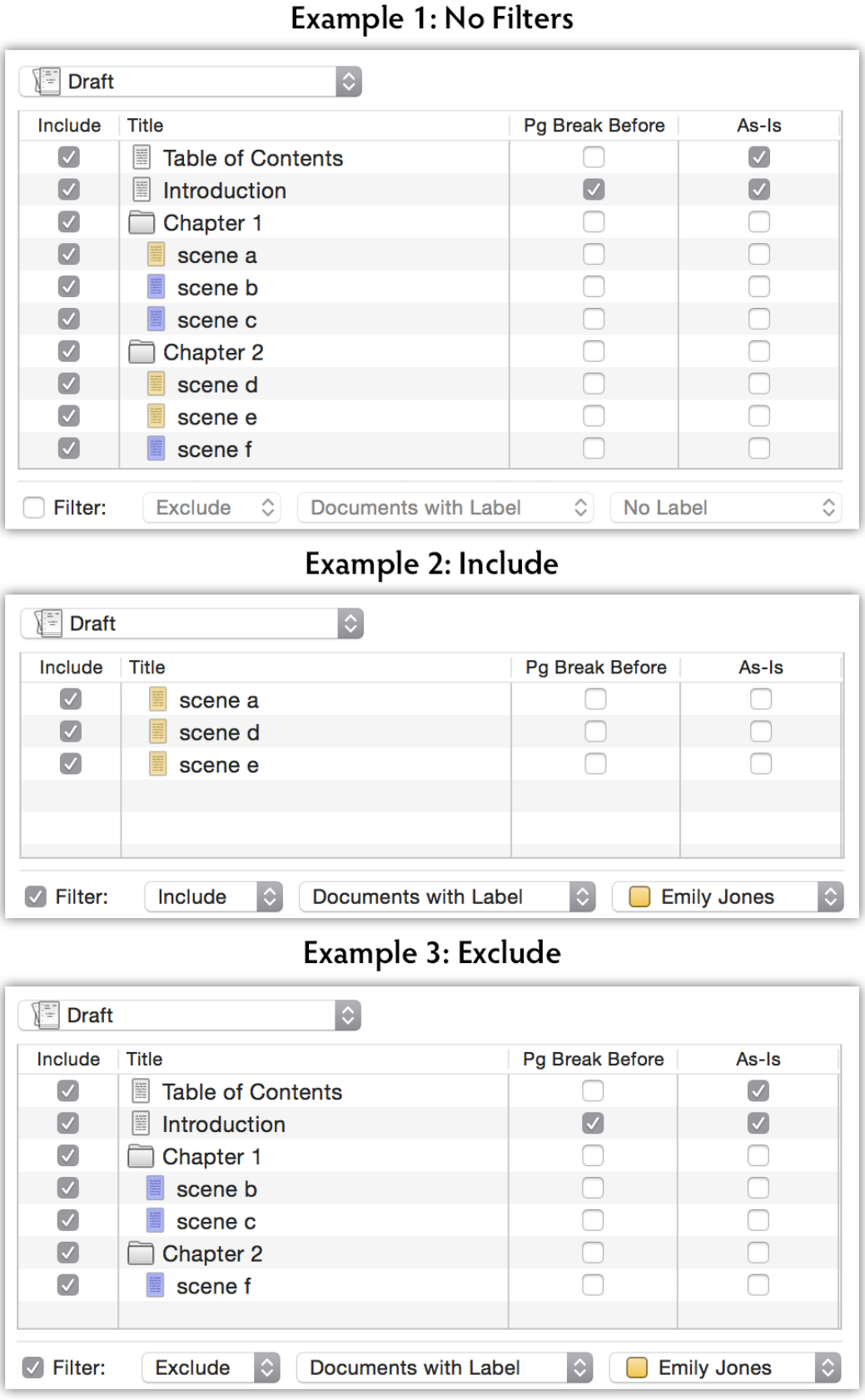

Figure 24.4: Sample filtering options

they will not be selected for compile unless they originally fell within the compile group scope.

Contrary to the method of selecting a Collection as your compile group, using a Collection as a filter will preserve any existing hierarchy within the original list. Of course, if the original list was itself the product of a Collection, there will be none. Thus, filtering by Collections can often produce a similar *content* result to selecting a Collection group, save for retaining outline structure.

The one other major difference between the two methods is based on the concept described above: filters can only modify the contents of the list they have been provided with. The items listed within a Collection will be used to match against the given compile group list, and they will never produce new results outside of the original list. So if you wish to include items from a Collection that do not exist in the original compile group, you will need to use that Collection for your compile group.

Collections are an effective way to arbitrarily focus your Draft for special purposes, without adjusting the "Include in Compile" checkbox for these items, but do note that this feature is secondary to the Include checkbox. The checkbox will always have priority over any filter settings. Filters adjust availability of an item in the list, they do not dictate the terms of compile beyond that point.

24.7.3 Front Matter

Front matter is traditionally defined as the material preceding the main content of a book, including everything from the title page to the preface, usually terminating with the first part or chapter. In print publishing, this often includes a different header and footer style, as well as Roman numerals for the page numbering.

As it becomes increasingly important to be able to deliver material in multiple formats, there is a rising necessity for swapping out front matter dynamically, depending on your compile format. You may need, from one single project, a PDF delivered to a Print on Demand service, an e-book published to Amazon or iTunes or a proofing instruction page for your editing team or a standard manuscript submission. In one, you might want a table of contents that uses page numbers, but in the e-book you won't have page numbers available so you wouldn't want those codes visible; you might not even want a table of contents at all. Using the Front Matter feature, near marking 4 (Figure 24.3), it is possible to quickly select between sets of files depending on what your current compile target is. If these folders are themselves contained within a folder called "Front Matter", Scrivener will prioritise their availability in the selection menu, by placing them at the top of the menu, regardless of where the folder is located in the Binder.

> **Want to see front matter in action?** The "Novel" project template provided with Scrivener demonstrates a possible setup with a group of individual front matter folders for use with corresponding publication outputs, such as standard submission or self-publication with e-book and PDF.

Selecting a folder as a front matter folder will transpose all of the contained items (and descendants) to the very top of the compile contents list without altering the actual order of these items in the Binder. This will happen no matter where the source folder is located (though to avoid confusion, you should never place the Front Matter folder inside the Draft folder itself).

In addition to swapping out sets for multiple outputs, the front matter feature can also be used with a few other compile settings:

- When publishing to the Mobi format, the compiler can set the point where the book will open initially to fall after the front matter contents. Read more in the Layouts compile option pane (section 24.13.3).
- For formats using page numbers, an alternative page numbering style, such as lowercase Roman numerals, can be used for all pages falling within the selected front matter folder, in Page Settings (subsection 24.22.3).
- Scrivener Links used within the front matter documents can opt out of having their prefix or suffix added to the readable link text, in Title Adjustments (section 24.12).

The Front Matter feature will be disabled (your settings will remain, but will be unused), when either of the following are being used as the compile group source:

- A subgroup of the Draft is selected *and* "Treat compile group as entire draft" is disabled.
- Compile groups based upon the current Binder selection.

24.8 Print Settings

Available to both the "Print" and "PDF" output formats, this pane gives you the choice between laying out your document using the publishing or proofing engine. These names have been chosen to reflect common usage, but are not meant to be precise. You may find the "proofing" engine to be perfectly adequate for producing a final version, or "publishing" better for proofing. The choice made here will impact the availability of options in the Layout (section 24.13), Footnotes/Comments (section 24.21) compile panes, and the First Page section layout settings in the Formatting (section 24.11.2) pane.

Publishing Based on an improved version of the system that Scrivener has used in the past, this method provides the most functionality for self-publishing, fiction or other simple texts that do not require footnotes. It will support such features as custom scene markers at the bottom of the page (where otherwise an empty line would be used), recto/verso placement for new sections, and section based header/footers.

Footnotes will be exported as endnotes when using this mode. Linking will be used to allow for easy navigation between the note and back to the reference marker in the main text, by clicking on the number.

Proofing Supports more sophisticated layouts, such as columnar layout, hyphenation and footnotes. This method will be better suited toward academic work, journal publishing and non-fiction—or if you prefer to see footnotes at the bottom of the page while proofing, rather than endnotes, as the Publishing choice provides. The primary negative to this engine is that it can exhibit faulty font usage on occasion, which is the primary reason for calling it a proof quality choice.

Links URL based hyperlinks can be underlined and coloured (in blue), with the following two options. If you wish to disable hyperlinks entirely, you may do so in the Transformations option pane (section 24.15).

24.9 Separators

Separators are used to automatically insert space, breaks, or custom symbols between the various elements in the list which will be compiled. The important aspect to realise is that by the time the compiler gets to this point, your outline is being considered as a flat list of files. It doesn't matter if a text file is four layers deep, if the next item in the list is a folder, it will still use the "Text and Folder Separator" rule.

Separators will not clean up any manually typed in separation that may exist between items in the editor text itself. For example, if you have items with a lot of empty lines in between them, the separators control will only establish if any *additional* separation should be added.

There are four primary types of possible combinations, each with its own individual rule. You can supply the same rule for all of them if you do not need any special differences between types. The types are:

- *Text separator*: inserted between any two adjacent text documents
- *Folder separator*: inserted between any two folders
- *Folder and text separator*: inserted at the top of any file that directly follows a folder

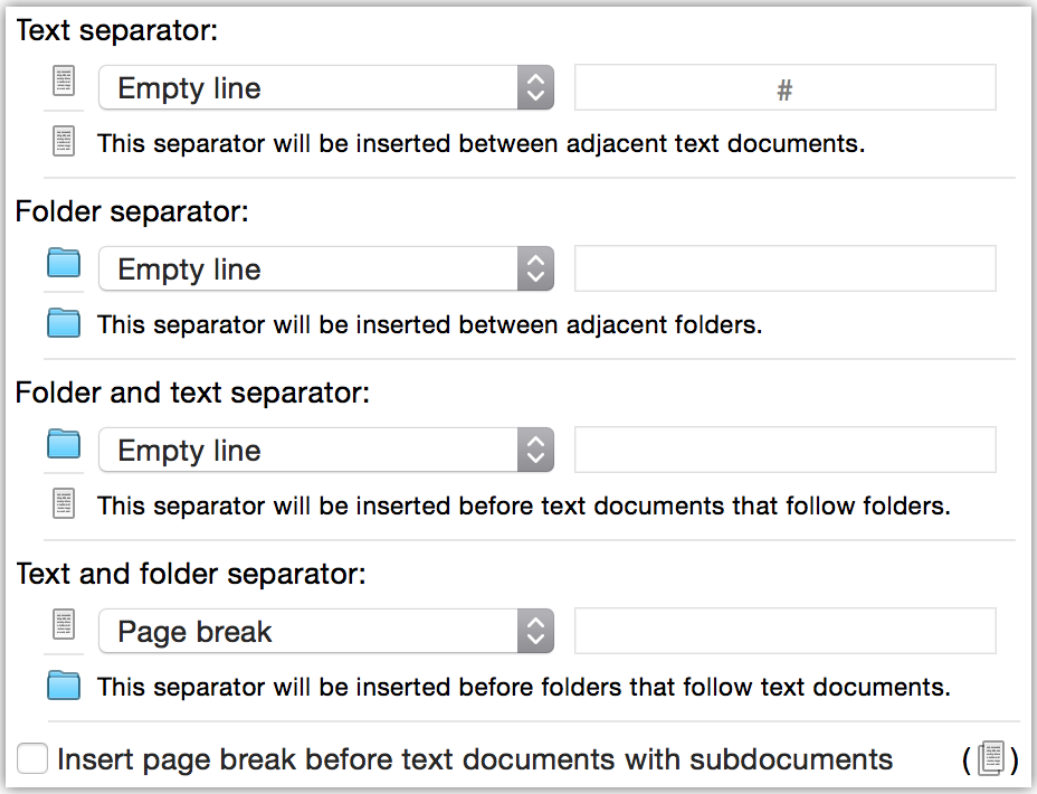
Text separator:
Empty line
#
This separator will be inserted between adjacent text documents.
Folder separator:
Empty line
This separator will be inserted between adjacent folders.
Folder and text separator:
Empty line
This separator will be inserted before text documents that follow folders.
Text and folder separator:
Page break
This separator will be inserted before folders that follow text documents.
Insert page break before text documents with subdocuments

Figure 24.5: Compile—Separators Pane

- *Text and folder separator*: inserted at the top of any folder that directly follows a text file

Each individual rule has four options, some of which change their behaviour depending on the compile format:

- *Single return*: a single paragraph break will be inserted, causing the final appearance to run from one document to the next with no visible "seam", and is the most minimal option you can choose
- *Empty line*: two paragraph breaks will be inserted, causing a visible space between the items. This method will trigger a non-indented paragraph following the break, if "Remove first paragraph indents: after empty lines and centred text" is in use (subsection 24.11.3).
- *Page break*: a page break code will be inserted, causing the following item's text to move to the next page of the manuscript

- *Custom*: Anything entered into the adjacent text field will be placed between the two items. Note that the text or symbols entered here will be buffered by a single return on each side, forcing it to fall on its own line. The value will be centre aligned. This method will trigger a non-indented paragraph following the break, as "Empty line" above behaves with the referenced option above. Image Placeholder Tags (subsection 15.5.4) can be used here, as a way of using a custom separator graphic between sections.

The following formats have different options and behaviours from those listed above:

Plain-text Files In the case of page breaks, the Unicode break character will be inserted. While support for this is not common, the character can still be useful as a search-and-replace anchor. TextEdit will treat this character appropriately when the file is viewed in Page Wrap mode.

HTML and WebArchive The "single return" and "empty line" options will be replaced by suitable alternatives to produce the same appearance of actual paragraph breaks.

Since page breaks have no meaning in Web publishing, a horizontal rule will be inserted. The option will be renamed to "Section break".

ePub and Kindle Since page breaks have no meaning in an e-book, the "Page Break" option will be changed to "Section Break". This will not only cause the e-reader to switch to a new virtual page (if it chooses to display things in such a way), but will also insert the name of the item with the section break into the automatically generated Table of Contents.

iBooks Author Chapters As with ePub and Kindle files, the "page break" option will be changed to "chapter break". This will determine where one .docx file is completed and the next one begins; the end result being a sequence of files, one for each chapter.

MultiMarkdown, and MMD HTML formats In most cases the "empty line" option should be used to retain clean paragraph breaks between blocks. The "single line" type will cause the last paragraph of one section to flow directly into the next paragraph of the next section.

Page breaks are replaced by "section breaks", which emit a series of hyphens on their own line. This will be converted to a horizontal rule using the HTML converter.

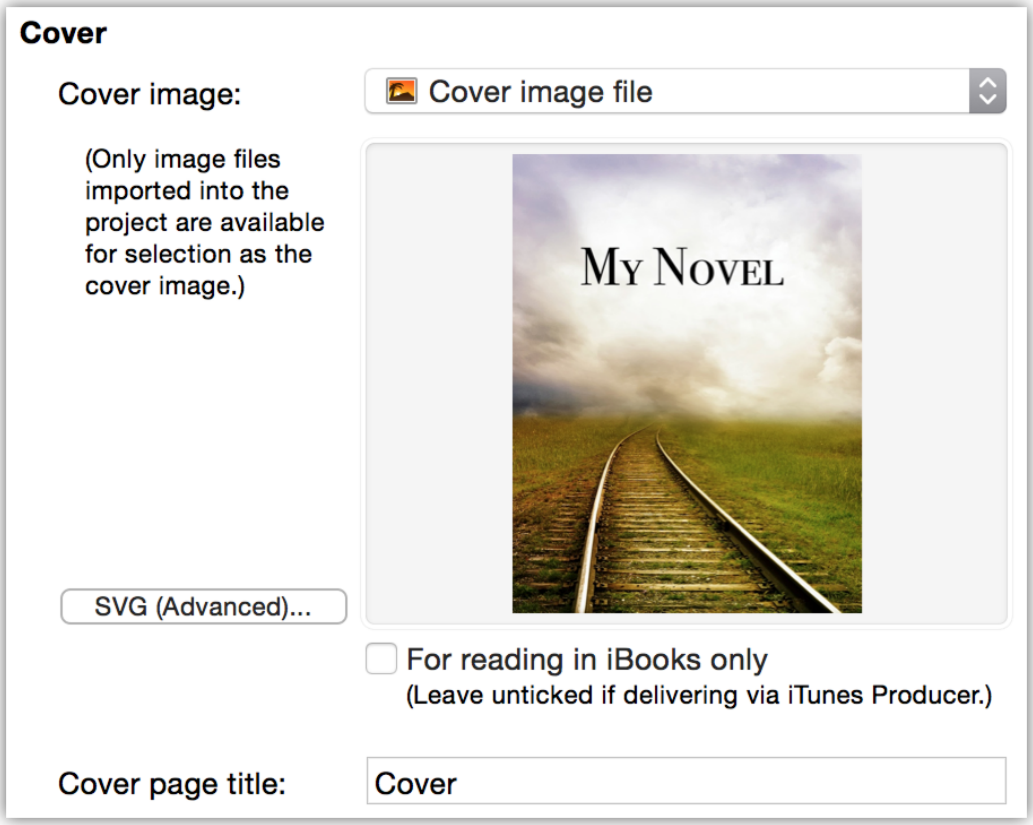
Cover
Cover image:
Cover image file
(Only image files imported into the project are available for selection as the cover image.)
My Novel
SVG (Advanced)...
For reading in iBooks only
(Leave unticked if delivering via iTunes Producer.)
Cover page title:
Cover

Figure 24.6: Compile—Cover & ToC

MultiMarkdown LaTeX format As with the other MMD formats, use of the "empty line" type will produce the best result in most cases.

The LaTeX code for a page break, \pagebreak, will be inserted for the "page break" option. Note that this will be enclosed in an HTML comment, which is the preferred method for passing LaTeX codes straight through the MMD system. This ability is disabled by default, and will need to be enabled by hand in the MMD sub-system (chapter 22).

The Separators pane is available to all formats except FCF and FDX.

24.10 Cover

Available only to the ePub and Kindle exporters, the Cover pane sets the cover and presentation material which will give your e-book a professional finish. The cover image will be placed at the very beginning of the e-book (before the table of contents), and for

ePub files, can optionally be included as the thumbnail image that will appear when the book is viewed on the "shelf" in the iTunes library and within iBooks mobile. Most e-book readers automatically use the cover image itself to provide their thumbnails.

Very likely you've been provided with images from a designer that was assigned to illustrate your work. They may not provide you with the file sizes necessary for e-books, since this is still a new market. Graphics for cover art should be in a standard RGB raster format, such as JPEG[3], and not a vector format, such as EPS, nor CMYK colour space which is designed for printing. For dimensions, there is no firm rule, but a size of roughly 800 pixels tall by 600 pixels wide at 72 DPI will display crisply on nearly all reading devices and computer screens with unnecessarily bloating the size of the e-book. If you are unsure of how to make these adjustments to the files you have been provided with, you should contact your graphic designer with these specifications so that they can deliver a quality version to you at the correct size.

In particular, watch out for very large files that were originally designed for full colour press. These will unnecessarily enlarge the size of your e-book, make it difficult to send out proof copies via e-mail, and could even cause e-book reader devices to run out of memory.

Cover image Cover images must be selected from resources within the current project. If you have a cover that you wish to use with your book, make sure you have imported it into the binder first. All images will be listed in this drop-down menu, and will be displayed in the grey preview area in the middle of the pane once selected.

Include iTunes cover art file for iBooks [ePub only] Once a cover image has been selected, this option will be made available. It will include additional information that iTunes needs to set the shelf cover art in iBooks. In most cases you should keep this enabled, but if the book fails to load properly in your reader, you could try disabling this. Do not use this option if you intend to publish using iTunes Producer.

Cover page title [ePub only] You should not supply the name of your book here. This is what will be used in the table of contents internally, to identify the cover page. The standard "Cover" has been provided as a default.

[3] At the time of this writing, JPG is the safest format to use. Not only is it the most compact, some reader devices do not support the PNG format.

24.10.1 SVG (ePub Only)

This is an advanced feature which will allow you to insert SVG code which can be used in conjunction with, or instead of, a bitmap image. If you have been provided with SVG artwork code by a designer, or have created one yourself, use the SVG button to access the sheet. The text area labelled, "SVG Code" is where you will paste the vector format code.

You will need to manually specify a default view box size. Get this information from your designer, the graphics program you used to create the SVG code, or from the XML itself. The actual size used is less important than maintaining the original aspect ratio. Incorrect aspect ratio values (the factor between height and width) will cause the graphic to become squashed.

You will need to research which SVG standard your target readers will be using. At the time of this writing, most readers are capable of understanding SVG 1.0 and SVG 1.1 (including SVG 1.1 Basic). When copying the XML code, make sure not to include the XML declaration and doctype lines. Only paste the SVG element and its contents. This will be the line starting with:

```
<svg version="1.1" id="Layer_1"...
```

...and typically going to the very end of the XML file. Pasting the entire XML file will likely produce an invalid e-book.

An iBooks compatible cover graphic will not be produced when using SVG covers.

24.11 Formatting

The Formatting pane is where you will design the look and structure of your document. It is separated into two main areas, both of which we will gradually introduce:

1. *Structure and Content Table*: A table for assigning content to the various Binder item types available, and the creation of level specific content and formatting rules.
2. *Formatting Editor*: A format editing and live preview area, which is linked with and displays the formatting rules for the selected row, above.

In the supplied example (Figure 24.7), the manuscript will print all folder titles (including any text within that folder item itself), and only text for file groups and files. It will also override the formatting to Courier 12pt, and adds extra information to the folder titles, one of which warns readers that they are holding a proof copy. All of this might seem like magic right now, even if it gives you a taste of what the compiler is capable of,

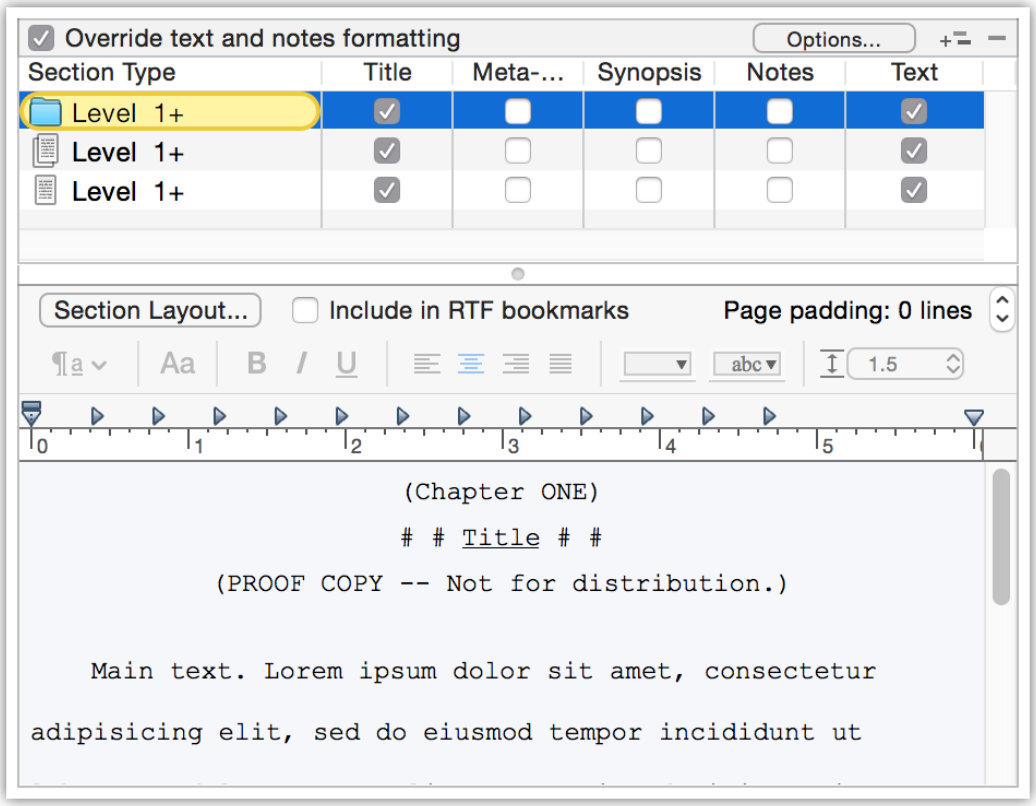

Figure 24.7: Compile—Formatting Pane

so let's take a look at the provided features. By the time you are done with this section, you should be able to see how all of this fits together, and in no time you'll be designing your own documents.

24.11.1 Structure and Content Table

The structure table initially has three rows corresponding to the three basic binder item types that can be found in a draft folder: folders, file groups (text files that have subdocuments), and text files[4]. For a moment ignore the "Level 1+" labels, we'll get to that in a moment. In the columns to the right of each type you will find a checkbox for each content element that can be optionally included by the compiler. Think of these as optional material you can include from your project, for *each* item in the Draft. As Scrivener pro-

[4]Some of the provided project templates might configure a number of formatting rows beyond the basic default three. A new blank project, however, will always just have three rows, one for each type. You might want to create a new blank project in order to follow along with this explanation.

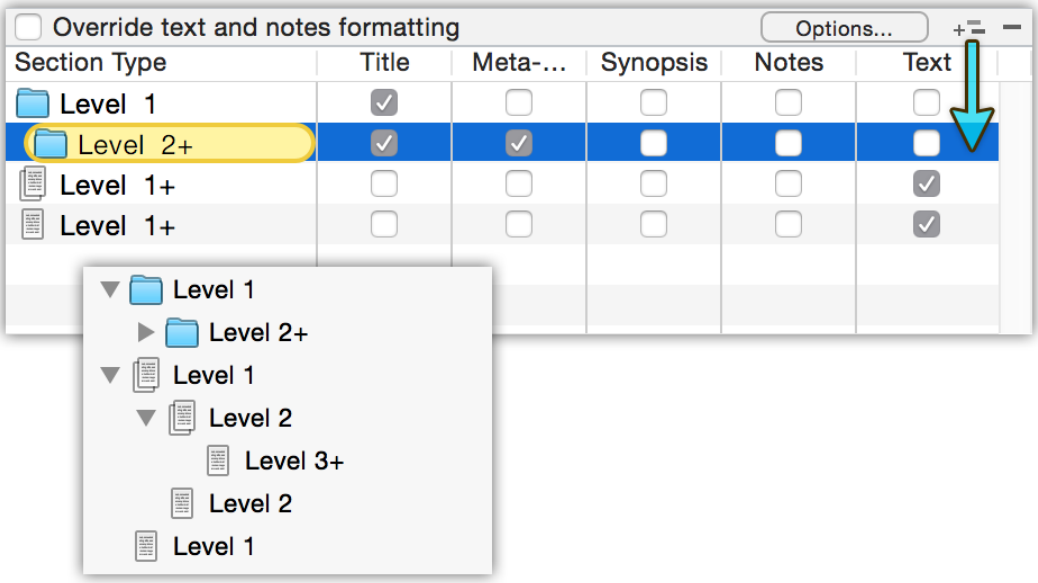

Figure 24.8: Example multi-level formatting rule.

ceeds down the list of items it has been instructed to compile, it will consult this table for instructions on how to handle it. If that item is a folder, it will look at the folder row and if "Title" and "Text" have been checked, will add the binder title for that item into the manuscript, followed by any text that is in that folder file. If the next item is a text file, it checks the text file row. If that row says only "Text" is exported, then it will print just the text content of that file into the manuscript.

Creative use of this feature can produce outlines (Titles only for all types), a document with only your notes and no manuscript text, chapter containers that export only their title, and so forth. Note that as you toggle elements on and off, you'll see a live preview of how they will be inserted into the compile, in the Formatting Editor below the table. We'll discuss how to change the look and feel of this in the next section.

Global Changes: You can enable (or disable) elements for all rows by Option-clicking on any of the checkboxes. If you want to apply settings to only certain rows, you can use the standard Copy and Paste functions to copy settings from one row and paste them into another, but be aware, this action will also paste any custom formatting settings from the lower pane as well.

Things get more interesting when these rules are applied to levels of hierarchy. This ability is best demonstrated with an example. Try selecting the Folder row in the table, and then click the "Add formatting level" button, indicated with an arrow (Figure 24.8). The label for the row that was previously selected will now be titled, "Level 1", and the

new indented folder row will be called "Level 2+". As you might guess, this means you now have *two* rules for handling folders. The first rule will apply to level one folders only, the second rule will apply to level two folders, and anything deeper in the hierarchy than that. To start, all configuration options from the original row will be carried over into the new row to help you get started in customising it. This holds true for formatting as well as the options we've already discussed.

What do these levels correspond with? They are matched with the indent level of items in your binder. An example binder structure has been inset in the example, with the items named to indicate how they correlate to "levels" in the Formatting pane. Items directly beneath the Draft are considered to be "Level 1", a child item beneath one of those is "Level 2" and so on. The way Scrivener's formatting engine works is by applying a format to the maximum depth in your draft outline. Inserting rules will thus only apply to those levels which it states in the label. In the provided example, I've added a checkbox to insert Meta-Data (creation and modification dates, labels, any custom meta-data and so forth) to any folders at level two or deeper. Folders directly descendant from the Draft will only print their titles in the compiled document, but all folders at level two or greater will also print their meta-data.

To make it easier to visualise what levels mean, in relation with your actual book outline, the Draft area in the Binder will highlight all of the items impacted by the currently selected Formatting pane rule. For example, if you have a folder "Level 1+" line selected, then *all* folders in the Binder will be highlighted.

To remove a level-based row, you can select it and click the - button to the right of the "Add formatting level" button.

There are other buttons and options in this table, but before we take a look at them, let's examine the formatting editor.

24.11.2 Formatting Editor

The formatting editor view is linked with the structure table above it. When you select a row in that table, you are also looking at the format for that row as well. Thus, each *type* of binder item can have its own formatting rules, and further *levels of depth* in the binder can have their own formatting rules. A very simple example of this would be different heading sizes depending on the level.

While added components, such as Title and Synopsis, can be formatted to your choosing, the main text area is special. To make any changes to the main text body formatting, the first thing you need to do is enable the formatting override. In the header for the structure table, click the checkbox labelled, **Override text and notes formatting**. When this option is engaged, your entire manuscript will be given a uniform appearance which can, if you choose, be radically different from how it looks in Scrivener. In a moment,

we'll discuss just how much can be reformatted, but for now assume the entire look and feel will be altered.

You will find a condensed Format Bar, a Ruler, some special options at the top, and a large text preview area. This cannot be typed in, it shows you what elements have been selected in the structure table, and how they will be formatted in the manuscript. You can otherwise treat it just like an ordinary editor. If you click in the "Title" area, you should see the formatting controls adjust to represent whatever formatting has been applied to the title. Try clicking the **Underline** button. Unlike with standard text, you are editing "areas" of text, so there is no need to select the entire title, you can just place your cursor anywhere within the title and click **Underline**. Refer to the documentation on using the The Format Bar (subsection 15.4.2) if you are unsure of what these various buttons do.

> **E-books and Fonts**: Unless you have purchased distribution rights for the font you've selected, fonts cannot be legally embedded into the e-book (and Scrivener is not capable of performing this action; it would need to be done after compiling). Consequently, the precise font family that you choose in this pane will probably not be the font you see in the e-book. In most cases, this is precisely what you want. The e-book reader hardware or software will provide fonts for the reader, and most will want to use their preferred font, size and styling if given the choice. The exception to this is that the font family *will* be declared in the formatting meta-data, so if that font is available on the device or computer, then it will in some cases be used. Again, this is up to the software itself as to whether or not it should override the reader's preferences with your font choice.

In addition to the standard controls detailed above, additional elements will be provided along the header bar of the formatting editor, if relevant to the particular format in use.

Include in RTF bookmarks When using the RTF export format, you will see an option next to the prefix and suffix called "Include in RTF Bookmarks". By default, every section in your binder will get an RTF bookmark, which will create a handy navigational reference in compliant word processors, and is also used to cross-reference links in Scrivener's table of contents feature (chapter 23), or your own Scrivener Links. It may not always be desirable to have bookmarks at every level of your outline, especially if you use Scrivener's outline feature to break down your book into small blocks. Simply uncheck this to remove the document type and level from the bookmarking feature. If you do this, you will be unable to link to this level and type of document in the ToC, or when using the linked <$p> feature for

cross-referencing by page number. Click the "Help" button in this sheet for some useful examples of title prefixes and suffixes.

HTML Elements When using a format based on HTML (which includes e-books), an element selector will appear next to the "Section Layout" control, using "Body text" by default—which is to say the standard paragraph element will be used. You may want to set your headings to use HTML style header numbers, where h1 is the highest level and h6 is the lowest. Each element within the formatting editor can be assigned an HTML element.

Page Padding Located on the right hand side of the formatting editor header bar, this controls how much padding will be added to the top of the page, in lines, when it happens to fall on a page break. Use this to offset your titles from the top of the printable margins if you prefer that look. This option will not be previewed in the Formatting pane, you will need to compile in order to see the results.

When using the plain-text output format, blank lines will be inserted wherever a page break would appear. Thus this setting can be used to visually space apart sections, even though in most text editors, page breaks are not possible. This option is not available to the MultiMarkdown formats, as this engine does not heed extraneous spacing.

Section Layout

The "Section Layout…" button provides options for adjusting the content boilerplate text around headings, heading formatting and letter case adjustments, as well as how the first page of the section will be composed if it is following a page break. These adjustments will only appear when you compile (though some will be previewed for you in the mock editor below the button). They will not be added to the binder items' titles or adjust the base text itself, and are thus useful for keeping clutter or awkward style choices (like all caps) out of the binder and editor. Rather than putting "Chapter <$n>:" into every single chapter folder in the binder, you can simply add that to the Prefix and be done with it, never having to worry about renumbering after an extensive restructuring again.

The Title Prefix and Suffix tab (Figure 24.9) handles the insertion of arbitrary text around the title of the binder item itself. If you typed "Test - " into the prefix field for folders, and compiled a draft with a folder named "Compiling with Prefixes", then the final output would be "Test - Compiling with Prefixes". In fact every folder included for compile would have "Test - " in front of its binder name. The suffix acts in a similar manner, only adding text to the end of the title. Changes made to the prefix and suffix fields can be viewed in the preview area, after dismissing the Section Layout window.

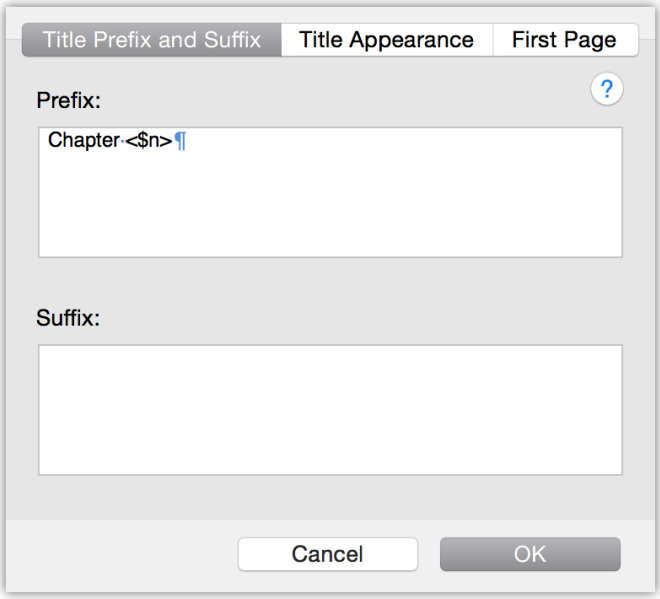

Figure 24.9: Title Prefix and Suffix tab

One important thing to consider is that the title prefix and suffix entries will be printed *even if the Title is disabled for that row*. This can prove very useful in some cases. It could for example let you use casual titles in the binder, and standard generic numbered titles in the final output. These are set specifically for each formatting rule established in the structure table. This means a folder on level 1 could use **Part** <$R>, while a folder on level 2 could use **Chapter** <$n>, and folders of level three and higher would use the hierarchical numbering provided by the <$hn> tag. Meanwhile text items could use something else entirely, or nothing at all.

Another creative example is to use one of the fields for something other than the title entirely. A proofing preset could be set up with a suffix that contains the document's meta-data on a new line beneath the title. You could use the Meta-Data checkbox, but if you wanted a more concise output with only a few key elements, this would be an easy way of doing so.

Using Graphics in the Title Area While you cannot insert graphics directly into these boxes, it is possible to insert a directive that will insert the graphic you desire when compiling (this will not be previewed). Read more about using Image Placeholder Tags (subsection 15.5.4).

Once added, formatting for the prefix and suffix can be handled independently from the title. If you wish to maintain a uniform look between the three components, use the mouse to select all of the visible title elements before styling. If the entire heading

will be output on a single line, then only character formatting may differ between the three. If however components are located on a new line (you can add carriage returns to the prefix and suffix box, to accomplish this), then each can have differing paragraph and ruler settings, too.

> **Using Distinct Formatting for Numbers** A common typesetting technique for chapter titles is to use a larger numeral. If you place the counter tag in the same field as the rest of the boilerplate text, you will be unable to size the number differently. Instead, place the boilerplate text (such as "Chapter ") into the prefix and the numeral tag in the suffix. Now you can control the font characteristics of the two components independently.

Tab characters and carriage returns can be added to these fields as well, and they have been set to show invisible characters in order to facilitate designing more complex or multi-line title adornments.

In both cases, no spaces will be inserted around the prefix or suffix. You will need to establish all spacing, if any, to be used in setting them apart from the main binder title.

Prefix Anything typed into this box will be printed before the relevant binder item's title, on that same line. If you intend to use this to insert the first part of a multi-line title, insert at least one carriage return after the boilerplate text in this box.

Suffix The contents of this box will be printed after the binder item's title, or if the binder item is not being printed, directly after the prefix. Consequently if you intend for this to display information on a line below the main title, insert at least one carriage return prior to typing anything in.

MultiMarkdown and Titles The title prefix and suffix will ordinarily be placed within the hash marks that Scrivener generates to indicate title depth. If you prefer, you can disable the "Place prefix/suffix inside hashes" options to allow text entry outside of the header line itself. To allow compatibility with the other compile formats, you will need to manually type in carriage returns to separate lines. Placing characters before or after the hashmarks on a title line will at best cause it to no longer render as a title correctly, and so hashes will be removed from either prefix or suffix. This makes for an easy way to override the default hash generation behaviour and supply your own markup around the title.

Title Appearance

In the **Section Layout...** dialogue, the Title Appearance tab (Figure 24.10) has three options for transforming the Title, Title Prefix, and Title Suffix, independently. These

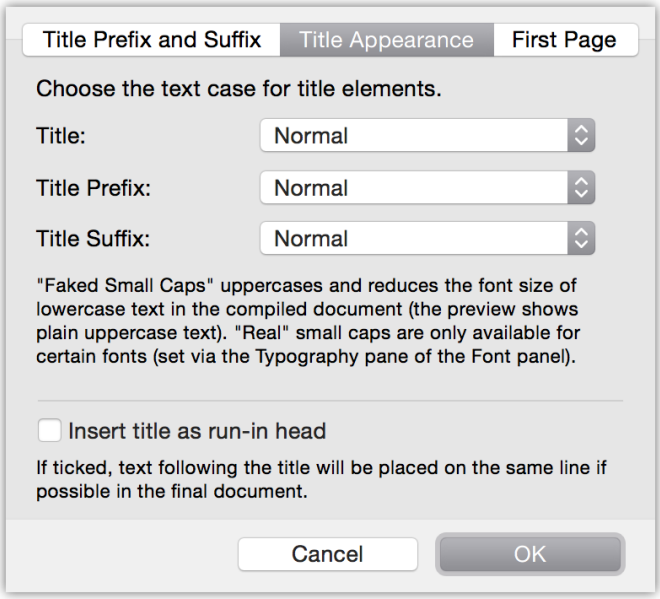
Title Prefix and Suffix
Title Appearance
First Page
Choose the text case for title elements.
Title:
Normal
Title Prefix:
Normal
Title Suffix:
Normal
"Faked Small Caps" uppercases and reduces the font size of lowercase text in the compiled document (the preview shows plain uppercase text). "Real" small caps are only available for certain fonts (set via the Typography pane of the Font panel).
Insert title as run-in head
If ticked, text following the title will be placed on the same line if possible in the final document.
Cancel
OK

Figure 24.10: Title Appearance tab

transformers adjust the letter-case appearance of the text while compiling, so you don't have to use uppercase titles in your Binder.

It also lets you use simulated small caps. True small caps are a function of a font's typographic settings, and few fonts have these capability built into them. Simulated small caps will adjust everything to uppercase, and then tweak the font size for each letter so that it appears to be a small cap font. Consequently, this option should not be used if you will need flexibility down the road to change that in a word processor, or need publication quality output for self-publication.

Note that for e-book publishing, faked small caps are more compatible as most e-readers lack small cap fonts.

Fake Small Caps cannot be previewed in the Formatting pane, though they will acquire an uppercase appearance to help indicate it has been changed. You will need to compile in order to see the results.

When using plain-text formats, the Faked Small Caps option will render the text in uppercase, instead. The setting will remain as-is, however, in case you switch back to a formatted output later on.

Insert title as run-in head When the title (or title suffix) is immediately followed by a standard text block (Main text, Notes, or Synopsis), it will be output in the first

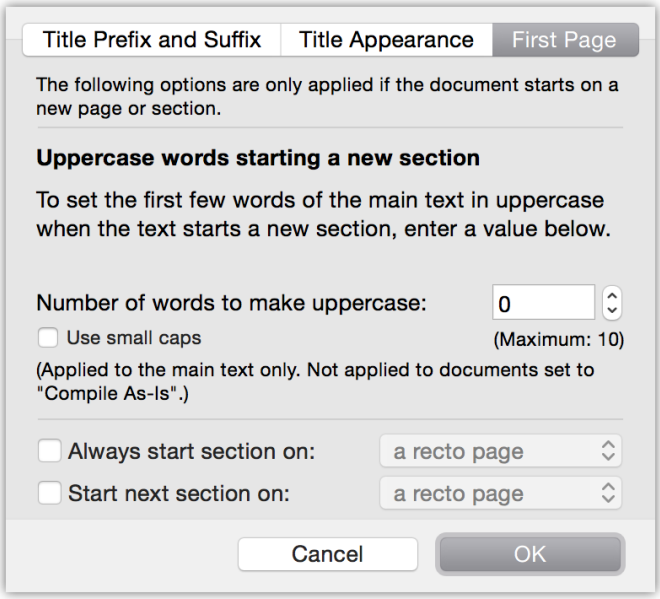
Title Prefix and Suffix
Title Appearance
First Page
The following options are only applied if the document starts on a new page or section.
Uppercase words starting a new section
To set the first few words of the main text in uppercase when the text starts a new section, enter a value below.
Number of words to make uppercase:
0
Use small caps
(Maximum: 10)
(Applied to the main text only. Not applied to documents set to "Compile As-Is".)
Always start section on:
a recto page
Start next section on:
a recto page
Cancel
OK

Figure 24.11: First Page tab

paragraph of that text. In the case of using a suffix on its own line, the suffix will be moved into the first paragraph, not the title.

When using run-in headings, the font and character attributes of the title will be used to style the title, but its ruler settings will be ignored in favour of the Main Text settings. A single space will be added after the title, so if you want to embellish it with an em-dash or some other sequence, use a Title Suffix.

This option will not be previewed in the Formatting pane, you will need to compile in order to see the results.

First Page

In addition to the structure and formatting of the heading itself, it is possible to affect the following text in the main body with a capitalisation option, so long as it falls on a new page, as well—when using PDF or Print—whether that new page should always fall on one side of the printed book or the other.

Uppercase words starting a new section If your formatting requirements are such that the first few words of the paragraph following a title need to be uppercase, then you can use the Text tab (Figure 24.11) of the **Section Layout...** dialogue to do so. This option will skip over any documents that have been set to Compile As-Is.

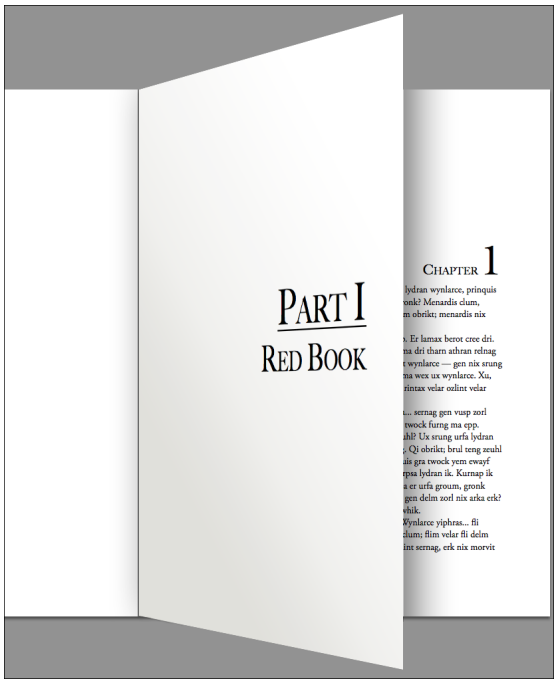

Figure 24.12: Example finished form output

You can also opt to use faked small-caps instead of all upper case, using the checkbox below this setting.

Always start section on The new section can be forced to always start on the verso (left) or recto (right) side of the book. This will in some cases cause an empty page to be inserted, in order to keep the chapter on the chosen side. This and the following option are only available when using PDF or Print as a "Compile For" option.

Start next section on If the *following* section also has a page break, this setting will control how it behaves.

Using these two options, you can set up common typesetting techniques (Figure 24.12), such as setting a "part" page to be displayed all by itself on the recto side, with the chapter page following it on the recto side as well, with a blank page in between them to do so. The example binder structures (Figure 24.13) demonstrate how this could be done with two different common ways of structuring a book in Scrivener. Example (a) uses smaller sectional files within chapter folders (highlighted in blue), which are themselves organised into part folders (highlighted in yellow). Example (b) has single-document chapter files (again in blue) organised into part folders.

To accomplish the desired output for example (a), one would need to:

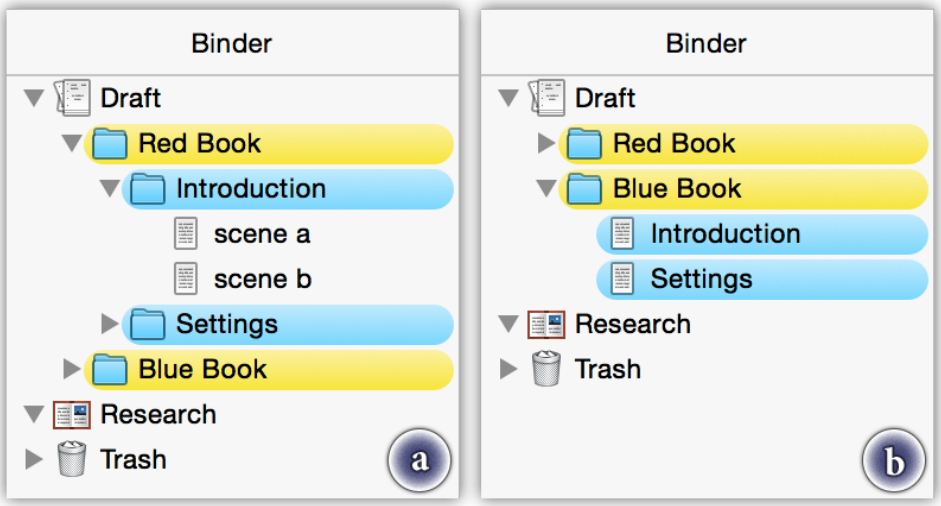

Figure 24.13: Two example Binder structures

1. Set their Separators (section 24.9) so that both the "Folder separator" and "Text and folder separator" are set to emit a page break
2. In the "First Page" settings for Level 1 folders (those folders directly indented beneath the Draft) *both* of the above options would be enabled and set to "a recto page"
3. Finally, a Level 2 folder row would be added with these options disabled (and thus stopping the cascade of the settings in the prior step to lower folders).

Example (b) would be similar:

1. Page break would need to be set for "Text separators" and "Text and folder separator", so that a sequence of files will generate page breaks
2. No change from the second step, above
3. Since folders are not used at any depth below level 1, it is not necessary to do anything further

Naturally, in both cases you would want to style the folder and text document formatting in a manner befitting their purpose. In both cases, the directive for a chapter to fall on the recto page following a part break would only be defined from the occurrence of a level 1 folder, followed by anything else that has a page break.

> **Is there a template for this?** With all of that explanation out of the way, you'll find a practical working template which utilises the sort of structure seen in example (a), in the *Novel (with Parts)* project template's default compile settings. You of course needn't stick by this structure, but hopefully you can see how the compile settings can be tuned to work with most any organisation you may require—the template just demonstrates one possible conclusion.

As with all other options in this panel, they can be applied per type and level, based on the selected row in the table above. Remember you can use copy & paste to move bulk settings between rows in the main formatting item type table. So it is is a good idea to set up sweeping changes first, then go in and tweak the individual types as necessary.

24.11.3 Special Formatting Options

Returning to the structure table, the **Options...** button in the table header bar will open a sheet which lets you set just how much of your custom formatting will actually be applied to the compiled manuscript, as well as a few other options.

These settings are global to the entire compilation, and cannot be set specifically per formatting rule.

> **Selective Opt-Out**: You can always set specific parts of the Draft to ignore all formatting, no matter what settings you choose in the compiler, by using the **Compile As-Is** Inspector option; which also can be strategically administrated from the Contents compile options pane. This will even ignore selected elements from the structure table, such as whether Title or Notes are checked off. Only the *Text* of such items will be output (even if it otherwise would not output text), and it will be output precisely as shown in the main editor. Absolutely no options from the Formatting pane will be used when outputting these sections.
>
> While there is no formatting to override in the MultiMarkdown workflow, this option will still inhibit the generation of automatic titling, even if the document would otherwise be titled by the compiler.

Insert subtitles between text elements When either Synopsis or Notes elements are enabled, Scrivener will insert a subtitle to help set apart these sections from the main text, which will itself also be titled. These titles can be formatted like everything else in the formatting editor in a unified fashion. You only need to edit the formatting of one subtitle to impact them all.

Place notes after main text The default is to place any Inspector notes above the main text for the item being compiled. When checked, notes will be placed below the main text area instead.

Custom Separator Centering Ignores Indents On by default, this will cause custom characters in the Separators compile option pane (section 24.9) to be truly centred on the page or column. With some layouts, you may wish for indent settings to impact the position of the custom separator so that it is offset from true centre. Disabling this option will cause the separator to use the main text indent settings for the formatting level in which it follows. For instance if a separator, "* * *" is set to appear between two text documents, then the indent for text documents will impact where the separator appears, horizontally.

Remove first paragraph indents Utilises the common typesetting practice of discarding the first-line indent for any paragraph following a header and/or section break. The calculation for this can be tuned with a set of options below the main checkbox.

- *On new pages only*: first-line indent will be stripped only when the item has been configured with a page break, either via a procedural declaration in the Separators pane (section 24.9), or individually for the item, via the "Page Break Before" toggle in the Inspector. Use this when you break down your work into pieces that are smaller than reader-accessible sections in your work.
- *At the start of each new document*: this is the default behaviour. Whenever a new document is encountered the first-line indent of the initial paragraph will be stripped out.
- *After empty lines and centered text*: unlike the above two options, this can trigger the removal of the first-line indent from even within an individual section. A full blank line (often used to denote scenes or section breaks), or the presence of any centre-aligned text (such as a scene separator like '#') will trigger it. This can be useful as a compromise between the above two options, where your outline is a casual representation of the reader-accessible structure, and might not strictly conform to section breaks or employ a mix thereof.

Text Override Formatting Options This section of the options panel lets you specify just how much formatting should be changed in the source documents. Note that if you use these, any respective styling you've done in the formatting editor will be ignored. For example, if you set the option to preserve line-spacing, then adding

double-spacing in the formatting editor will do nothing. Access to this section will be disabled unless "Override text and notes formatting" is enabled.

- *Override font only*: If all you want to change is the base font family, checking off this box will disable the other options and inhibit the compiler from changing alignment and ruler styles.
- *Preserve alignment*: Left, centre, right, and full justification settings will be preserved throughout the manuscript. The sub-option, "Centered Text Only" will cause the engine to only preserve centre aligned text (such as manually typed in scene breaks) while forcing everything else to use the relevant formatting options.
- *Preserve tabs and indents*: Actual tab characters will always be preserved. This option merely inhibits the compiler from changing the ruler based tab stops and indent markers. If you use blockquote indenting in your book and wish to retain that formatting globally, use this option.
- *Preserve line spacing*: All of the spacing options, including paragraph spacing, will be left alone.

Preserve Formatting only preserves Adjusts how the Format ▸ Formatting ▸ Preserve Formattingfeature works (subsection 15.4.6). The default settings will block all compile attempts to change formatting within the range of text within the block.

Merely by enabling this feature, you automatically opt-out of special-purpose fonts within the Preserve Formatting block. Using these options, you can selectively weaken that feature to allow some types of formatting to be overridden anyway:

- *Alignment*: Left, centre, right, and full justification will be passed through to the preserved block.
- *Tabs and indents*: Tab stops and ruler indents will not be preserved in the block.
- *Line spacing*: preserves any settings from the spacing configuration. This not only includes the most obvious spacing between the lines themselves, but also any offset between the paragraphs themselves.
- *Font size*: The size of the font will be made uniform with the text around it, but not the font family.

Handling Blockquotes: Blockquoting presents a unique typesetting situation in that you often need a combination of some settings retained from the Preserve Formatting

> block, but not all of them. A common configuration would be to enable the restriction checkbox, and then de-select only alignment, as blockquotes typically are set in a smaller face, using single line-spacing, and a left offset indent from the margin.

24.11.4 Managing Formats at a Higher Level

With so many options available in regards to the specific formatting itself, within the mock editor, it is useful to have management tools for moving these formats around between rows. Say you've set up a meticulously designed formatting template for second level file stacks, and realise that with the exception of the title prefix, you want everything else to be applied to second level text files too. You can use standard copy and paste commands to transfer *all* settings from one row to another.

Copy and paste portions of formatting If all you want to do is copy the formatting you've applied from one element to another in a different area (even in a different row), click anywhere within the sample element you wish to copy from and then use the standard formatting copy and paste tools for doing so:

- *Copy ruler*: ^⌘-C
- *Paste ruler*: ^⌘-V
- *Copy character attributes*: ⌥⌘-C
- *Paste character attributes*: ⌥⌘-V

The following two tools are row level actions, you must have a row in the structural table selected in order for them to work.

Copy and Paste All Settings Between Rows The keyboard shortcut commands for Copy (⌘-C) and Paste (⌘-V) can be used to bulk transfer all settings for both the structural table (whether title, notes, etc. export) and the entire formatting pane below. A common tactic is to set up one row just the way you want, then paste it to all of the others and tweak them as necessary.

Setting the font for all elements To set the base font for all elements within the formatting editor, select the row for the type and level you wish to edit and use ⌘-T to bring up the font palette. Any changes made here will impact every single element uniformly, so save this method for the very beginning of the customisation process. You will lose variant, size differences, and other characteristics that have already been applied to various elements.

When using this method, even if the main text area is not being overridden, the font will change. This does not mean that your draft will be reformatted (unless the override option is selected later on, of course).

24.12 Title Adjustments

This is a special option pane in that it will only appear when Formatting rules (section 24.11.2) have, at least once, declared that a title be adorned with a prefix or suffix. These options further define how adding these title settings should function within the context of manuscript, including how other items even *referring* to the documents should be treated.

For example, if you have instructed folders to be prefixed with "Chapter <$n>", causing that to become a part of the visible and effective title at the reader level, you can ensure that all cross-references pointing to those folders (via Scrivener Links) are likewise modified.

Do not add title prefix and suffix to documents... Here you can set specific documents to not use any prefix and suffix settings, even if they are otherwise scheduled to do so. Common examples would be the preface, acknowledgements, and appendices. Note that while the Inspector option, "Compile As-Is" can also be used to inhibit title prefix and suffix on a per-document basis, doing so comes at the expense of dropping the title itself and forcing the export of the main text body for that item.

You can also choose to omit prefix and suffix treatment to *all* items contained within the selected Front Matter (subsection 24.7.3) folder, if you are using this feature. This would often pertain to such items as the title page, table of contents, and so forth.

Update titles in Scrivener links with prefix and suffix settings When Scrivener links directly reference a binder item title, the compiler will be instructed to update the text of the link to match the final, altered, title. Since the prefix and suffix field can contain special whitespace characters, such as tabs and carriage returns, these will be stripped out of the result in order to keep the reference inline. Consequently you may wish to provide a separator using the **Title prefix separator for inline links**. The value placed into this field will be inserted between the title and the prefix, in replacement of any whitespace used in the title prefix. Click the help button, located nearby, for some practical examples of usage.

Do not include title suffixes in updated links will restrict this renaming option to only use the prefix value.

Read more about linking in When Scrivener Links can be Compiled (subsection 9.5.7).

24.13 Layout

This is where general decisions about the overall layout of the document can be made, and as such it presents an array of different options depending on the output format. It is also where most of the book level presentation decisions (such as table of content look and feel) for e-books can be set. This panel is available to all formats except Final Draft.

End-of-Text Marker This option is available to all export formats. It will place the provided text at the very end of the compiled file, signifying the end of the document. Some submission formats require a special punctuation sequence at the end, and this option can be used to keep that sequence outside of the main working area in the editor.

Do not add page padding to the first document The "Page Padding" option, set up in the Formatting compile pane (subsection 24.11.2), normally inserts a blank area pushing the content of the page down from the top margin wherever a page break is encountered, or a section break, if the format uses these instead. If the first document in the draft would ordinarily create a page break, this option can be set to disable this behaviour. It will not remove the page break, only the offset padding.

24.13.1 For PDF and Print

One extra option is available when printing or using the PDF file format and the Publishing layout engine (section 24.8).

Empty Lines Across Page Breaks Enable "Replace empty line separators that fall across pages", and then supply a custom separator backup. This will be used only when an "Empty Line" Separator (section 24.9) is scheduled to be used, and that line would otherwise be hidden by the page change. This is a common typesetting convention for making sure that separations between scenes are indicated at all times.

24.13.2 For Rich Text and Proofing

With the word processor formats and PDF/Print set to the Proofing layout engine (section 24.8), this pane allows for some advanced page layout options, such as widow and orphan protection where applicable, hyphenation, and column based layout.

Use hyphenation By default, hyphenation will not be used. When full justification is in use, this option can substantially improve the readability and appearance of the compiled result. If you intend to use a word processor to handle the compiled file, it is often better to let it handle hyphenation, instead.

Avoid widows and orphans When used with a compatible word processor, this will enable widow and orphan[5] protection for your paragraphs.

Columns Columns will reformat your exported manuscript into a specified number of vertical columns. As with all compile features, this will have no impact on your visual draft, or while writing and editing, and only modify the appearance of the final product once exported. To enable the use of columns, check the "Use columns" box.

- *Start columns*: three choices are provided for where columnar format should start in the document:
- *On first page*: start columnar layout immediately
- *After first document*: useful for title pages or the abstract block in many style guides used by the sciences
- *After front matter*: if front matter is selected, it will be entirely displayed in single column mode.
- *Number of columns*: the number of columns on each page can be adjusted here. Any value between two and four can be selected
- *Space between columns*: you may also adjust how much padding will be used to space the columns apart from one another, from 1/8th of an inch to 3/4.

24.13.3 For E-books

This describes additional options available to the ePub and Kindle .mobi export formats, which impact the presentation and functionality of your e-book, such as cross-referenced table of contents and script formatting.

Additional Text Layout Options

Do not add page padding to the first document When padding has been added via the Formatting compile option pane (subsection 24.11.2) to offset the start of a

[5] Widows are remnant lines where the paragraph breaks across the page, resulting in only a few words after turning the page. Orphans are the opposite, where the paragraph begins so low down on the page that only the first line can be read before a page flip is required. This option will strive to reduce instances of this by moving paragraphs from one page to another to keep the text as cohesive as possible.

section from the top of the display (or page), you can optionally suppress this behaviour for the first section of the book (often chapter 1) with this option.

Downsize and resample inline images to visible size To cut down on the overall size of your e-book, you may wish to have Scrivener resample the graphics when compiling. Note that the images will always appear to be the size you have set them to in the editor, or the size they appear as naturally. Ordinarily, this is done by setting the visible size of the picture without changing its underlying pixels. This option will resize the graphics for the compiled version *only*. It is not destructive; your images in the project will remain untouched, as is the case for nearly every function in the compiler.

Include standard Adobe Digital Editions page template Adobe Digital Editions[6] uses a special proprietary file (page-template.xpgt) for setting the type and controlling layout look. Without this file, the appearance of your e-book may suffer in ADE and devices that use that technology, such as Sony, Nook and Kobo readers. This file is not considered part of the official ePub specifications, however, and can conflict with some validation and publishing services, most notably Apple's bookstore. If you do not intend to distribute through the bookstore, it would be a good idea to enable this option so that the book will appear as you intend it to on a wide variety of systems. This option is only available for ePub output.

Book begins after front matter For this feature to work, the Front Matter feature must be enabled (subsection 24.7.3). When active, the book will initially open at the first section that follows the material added as front matter. This can be useful if the front matter you are inserting with the feature is not essential for the reader to see, and you wish for them to start right at chapter one. Not all reader devices or software will comply with this setting. When unchecked, the book will open at the start of the entire book, and so is thus useful when the front matter contains text that is important for the reader to go through, such as a preface.

You can also manually supply a starting point by typing in the placeholder, <$ebook_start> into the section you wish the reader to start at. The placeholder will be ignored if this checkbox is active.

This feature is only available for Kindle Mobi output. The ePub format does not contain any capability for setting where the reader should begin when they first open the book on their device.

[6] http://www.adobe.com/products/digital-editions.html

Table of Contents

Table of contents are often handled by the device using a special interface which lets the reader jump directly to a designated spot in the book like a hyperlink. You will rarely need to create your own table of contents by hand. Scrivener will generate a table of contents based on items which have a section break assigned to them. Section breaks are generated in two fashions: either procedurally in the Separators compile pane (by default, folders following files automatically get one), or ones you have set up manually, using the "Page Break Before" Inspector flag.

Use flat list of contents in navigation controls (NCX) By default, the TOC list will be nested according to the depth of the items in the binder. To produce a flat list of items instead, enable this option. This only impacts the "software" ToC, not the HTML formatted ToC. On a Kindle, this list is not provided to the reader in the software, however the Kindle *does* use the NCX ToC to generate the dots that it places in the navigation progress bar at the bottom of the book page. These dots are used for navigation within the book via the joystick control, and they only use the top level of ToC entries to generate dots. Consequently, if you want dots for items of greater depth in your ToC, you will want to set this to flat list.

Generate HTML table of contents In addition to a "software" ToC, which can be accessed in readers that support the feature, you can also automatically generate a ToC into the text of the book itself. In most cases you will want to leave this on for increased compatibility. Not every e-book reader can access the "software" ToC, and disabling the HTML ToC will leave readers incapable of navigating within your book. For the Mobi format, Amazon will reject submissions that do not include an HTML ToC.

Center body text of HTML table of contents Will centre all of the titles rather than left aligning them.

HTML table of contents title If you have created your own table of contents page using Scrivener Links (also check HTML Options (section 24.16) and make sure they are enabled) you can specify the document title (as seen in the binder) for that file, here. This sets a special flag so it can be navigated to with the reader's navigation features. It will otherwise appear as an ordinary part of the book.

Alternatively, if you are happy with the automatically generated HTML ToC, but would simply like to move it to another location in your book, you can create a document in the binder using the name specified here for its title, and type <$toc> into it.

Screenplay and Script Formatting

If your e-book is composed of, or contains any script formatting, you should enable the "E-Book contains script formatting" option. Scrivener will insert special formatting rules which match the script format settings you are using in that project, and typeset these elements appropriately.

In most cases, the default styling will be acceptable, but with some custom formats that have difficult to emulate styles, you may need to customise the rules which determine the final appearance in the e-book. Knowledge of the Cascading Style Sheet[7] syntax will be required to effectively customise the look and feel.

Click the **Customize CSS...** button, and you will be presented with a text field where you can edit the CSS. Each script element will be provided with a separate class. Element names which are composed of two or more words will use the so-called "camelCase" which involves removing all spacing and capitalising each new word but not the first one. An example would be "Scene Heading", which would map to a class name of "sceneHeading". Single-word element names are simply addressed as a lowercase word, like "transition". All classes should be assigned to the **p** element.

Even though the formatting uses CSS, you should not expect to be able to tap into the full power of this syntax. Designing an e-book is a far cry from designing a web page. Many CSS commands simply won't do anything at all in most display software.

24.14 Script Settings

This set of options lets you configure a few details which impact how a script will be formatted. Most of the Script Settings options are only available to the Final Draft (FDX) format. However a few are also available to the Fountain plain-text screenwriting format. These options will be marked where applicable.

First document is title page If the first document in your current compile group is a special title page with no script formatting, leave this option enabled. It will be assigned to Final Draft's title page window, and kept separate from the script itself. If you've noticed the first page of your script seems to be missing, and you aren't using a title page, make sure this option is disabled.

Break dialogue and action at sentences Use this option to adhere to the standard of keeping action and dialogue sentences together, rather than breaking them up between pages. If a sentence would have ordinarily been split, it will instead be moved entirely to the following page.

[7] http://www.w3.org/TR/2011/REC-CSS2-20110607/

Include document titles as scene titles Available to Fountain as "section headers".

Enabled by default, the names of draft items as listed in the binder will be used as scene titles in Final Draft, or as section headers in Fountain. When Scrivener imports from either FDX or Fountain (via the File ► Import ► Import & Split... menu command), these will be used to generate binder item names.

The Fountain format has an additional option, **Section headers use binder indent levels**. Fountain section headers can optionally express depth, though this is not used in any way beyond the purely visual, in the text file.

Include document synopses as scene summaries The content of the synopsis field will be used to populate information into the "scene summaries" feature in Final Draft.

Include revision colors from Preferences Your preferred revision colours, which can be set in the Appearance preference tab, will be supplied to Final Draft's revision palette, maintaining a consistent revision system between the two applications.

Use default Final Draft screenplay elements By default, when using the "screenplay" script format, Scrivener's output will match that of the industry standards used by Final Draft. However if you've made changes for your own aesthetic tastes, or are unsure of the formatting in general, you can check this option off to remove Scrivener's formatting instructions and have Final Draft handle all of the formatting. This could result in a compile that looks different from what you've been writing, but in most cases there will be no visible change. Naturally, if you *require* a script format that doesn't conform to the standard screenplay, make this this option is unchecked, or you will lose all of your custom formatting in the compile (Scrivener's copy will of course remain untouched).

Include footnotes as script notes Inline and linked footnotes will be converted to Final Draft's script notes feature. Footnotes will be removed from the compiled document if this option is not enabled.

Include comments and annotations as script notes Inline annotations and linked comments will be converted to Final Draft's script notes feature. All comments will be removed from the compiled document if this option is not enabled.

Fonts The final two options allow you to set fonts for the indicated features within Final Draft. Do note that if you intend to share the FDX file with Windows colleagues, you may wish to change the default Summaries font to Helvetica, Arial, or something else that is commonly available.

24.15 Transformations

The Transformations pane provides text and formatting conversions (where applicable). Since many of the available formats have their own unique strengths and weaknesses, not all of the options listed will be available for every format.

24.15.1 Plain Text Conversions

These are useful conversion tools for translating special characters and some ruler styles to plain-text variants. These options are made available to the rich text formats as well.

Straighten smart quotes Typographic quotes will be substituted with straightened variants. In versions of Mac OS X Leopard and greater, the system language settings will be used to determine the proper substitutions.

Convert em-dashes to double hyphens The '—' character will be converted to --. If this option does not appear to be working, ensure that you have used *em* dashes (⇧⌥-) and not *en* dashes (⌥-) in your manuscript. The MultiMarkdown formats will convert em-dashes to triple hyphens instead, as the double hyphen is used to input an en-dash.

Convert ellipses to triple periods The '…' character will be converted to three full stops.

Delete struck-through text Any text that has had a strikethrough applied to it will be removed from the compiled file when this option is enabled, allowing a form of "soft-deletion" while editing.

Do not convert auto-number tags to numbers Scrivener's token counter system will be disabled (the tokens will remain visible). Note this only impacts the counter system. Other document tokens and replacements will still be expanded.

Remove trailing whitespace from documents Cleans up each document so that any extraneous carriage returns at the end of the section will be stripped out.

Convert multiple spaces to single spaces Cleans up instances where the original typing has two or more space characters after a full stop.

Convert to plain text Actual spaces and carriage returns will be inserted into the document to emulate indents and paragraph spacing (line spacing will be ignored) and even margins. This function determines spacing by rounding up the supplied value. If the base font is 12pt, and paragraph spacing is set to 28pt (factor of 2.3) then two carriage returns will be inserted between paragraphs. If applicable, this feature is

compounded with any overwritten formatting, as established in the Formatting compile option pane (section 24.11).

- *Paragraph spacing*: add spacing in conformance to any before or after paragraph spacing in styled text.
- *Paragraph spacing and indents*: in addition to the above, spaces will be inserted wherever lines have been indented. This works in a similar fashion, where spacing is rounded up. If the indent is roughly equivalent to three spaces, then three spaces will be inserted. This only simulates first-line indenting. Block quotes and other effects such as hanging indents cannot be simulated in this mode; use the following if you require that level of emulation.
- *All whitespace*: with the exception of full justification, this mode will attempt to faithfully preserve *all* whitespace, including alignment, right-indent offsetting, block indentation, hanging indents, and so forth.
- *All whitespace (add a one inch margin)*: in addition to all of the above, this mode also adds 10 spaces to the left of every line. Naturally, the actual size of this space will differ depending on the text editor you open the file in, and the font being used to represent the plain-text document.

This option is not relevant to the FDX and FCF formats, and will be removed from the pane when they are selected.

MultiMarkdown and Spacing: The "Convert to plain text" tool can prove to be of great benefit when working with the MultiMarkdown format, or indeed any plain-text marking system such as LaTeX, where the original source document was typed according to standard word processor conventions of only using a single paragraph break between paragraphs and using pseudo-spacing between the paragraphs, or indents, to achieve visual separation. Since many plain-text markup formats require a clean break between paragraphs, this option can automatically convert such a word-processor style document to a form that plain-text systems can work with.

Most of these formats do not recognise or use greater quantities of space than one, for instance five empty lines between two paragraphs is the same as one empty line, in MMD. Thus, it is safe to use this option even when a mix of spacing conventions have been used, as any overage in the correctly typed portions will not impact the output.

In most cases you will not want to emulate indents with MMD, unless they have only ever been used to indicate verse or code blocks. If paragraphs have been indented for aesthetic preference in the editor, then enabling indent conversion will turn every single paragraph in your compiled document into a code block.

24.15.2 Rich Text Conversions

This section provides options for automatically converting a few rich text conventions from one standard to another. They are only made available to the compile formats which support styled text.

Convert italics to underlines If the submission process requires underscoring to be used instead of italics, this feature will let you write in italics but produce a properly underscored manuscript.

This option is not available to any formats which produce a plain-text document.

Convert underlines to italics Use when you have produced a document with underlines, but need an italic version for compile.

This option is not available to any formats which produce a plain-text document.

Convert Markdown to bold and italics Text that has been formatted using simple Markdown[8] for strong emphasis and emphasis will be converted to bold and italics respectively. Note that if italics are being converted to underlines as well, then the end result will be underlines. *single asterisks*, **double asterisks** and _underscore characters_ are acceptable, as well as combinations of them.

Remove Highlighting Text highlights will be stripped from the compiled document in all cases, save for when highlights are used to indicate other features. Notably, when compiling to RTFD, Scrivener will use highlighting to indicate linked notation ranges. This setting will not impact this behaviour.

Remove text color Likewise as above, this option will strip all custom colouring from the compiled document and force all text to black, without exception.

Convert linked images to embedded images Available with the RTFD format. If you have used linked images in your project text, use of this function will find the original resources and embed them directly into the compiled document. You will nearly always want to use this option when producing a file that will be sent to other people or viewed from more than one computer. Since the other available formats do not support linking, they will always import and embed any linked resources during compilation.

See also: Linked Inline Images (subsection 15.5.3).

[8] http://daringfireball.net/projects/markdown/

Remove all hyperlinks Available to all word processor formats as well as PDF and Print, this will strip out all hyperlinks from the produced document. With some styles, this may be preferable where the URL is printed literally in the text as a citation.

24.16 HTML Settings

HTML Options control a few specific features in relation to the Web Page (but not .webarchive), ePub and .mobi formats. If you wish to adjust the HTML DocType specifications for exported HTML files, the technical details for how CSS is included (or not), and whether or not most ruler styles will be converted to style, see the Import & Export preference tab (section B.10).

Convert Scrivener links to HTML links When enabled, Scrivener will insert the necessary HTML framework to allow for links to cross-reference to internal parts of the Scrivener outline. If the linked element is outside of the scope of the current compile group, the link will be removed from the compiled document, rather than produce a broken link.

Footnotes will always use internal links to cross-reference from the marker to the footnote text.

Underline links By default, Scrivener links are styled so that they are *not* underlined in the web page. If you wish underlining to be retained, use this option to disable the stylesheet.

The following two options are only available to Web Page compilation.

Treat "Preserve Formatting" blocks as raw HTML A useful option when writing specifically for the purpose of web publishing. The Preserve Formatting tool can be used to output any raw HTML that you type into the editor, rather than its typical role of preserving formatting.

Use centered table to restrict body text width The entire document will be enclosed in a div which will keep the width of the document within a certain limit and centred in the browser. The precise width can be specified in the provided text field, in pixels. The default value of 600 pixels is optimised for a character size of roughly 12 points, which will maintain a conventional line length of 10–12 English (or similar) words for maximum readability.

24.17 PDF Settings

Applicable to only the PDF output format when the Publishing layout engine (section 24.8) has been selected.

Generate PDF outline Assembles a nested list of items that will appear in many PDF readers which support a content tree[9]. It *will not* generate a visible table of contents in the printable work itself, only in the menus and sidebars of applications which utilise PDF content lists. For visual table of contents, read the chapter on quickly making your own (chapter 23).

For an item to appear listed in the PDF contents, it must be provided with a visible title in the compile Formatting option pane (section 24.11). The title can either be the binder item's name, or a generic title generated by the prefix/suffix settings applied to items of its kind and depth. Please read the referenced section for more details on how to set up titling, if need be.

Underline internal links If you intend to print the PDF at any point, you will probably not want to colour or underline internal cross-reference style links, so this feature has been disabled as a default. For PDFs which are intended to be used purely digitally, underlining and/or colouring links is an important and valuable way to communicate links to the reader.

Color internal links Enables internal link coloring. Hyperlinks to URLs will always be coloured in blue, as is the standard. You can select a different colour (the default uses the same green used in this manual) for internal cross-reference style links.

24.18 Replacements

Replacements are a way for you to set up your own substitutions, which work in a manner very similar to Search and Replace, though without changing the underlying documents in Scrivener.

A table with five columns has been provided:

- *Replace*: the text which you wish to instruct the compiler to look for.
- *With*: whatever has been supplied in the *Replace* column will be replaced with what you type into this column.

[9] In Preview.app and Skim, you'll find the content list in View ▸ Table of Contents.

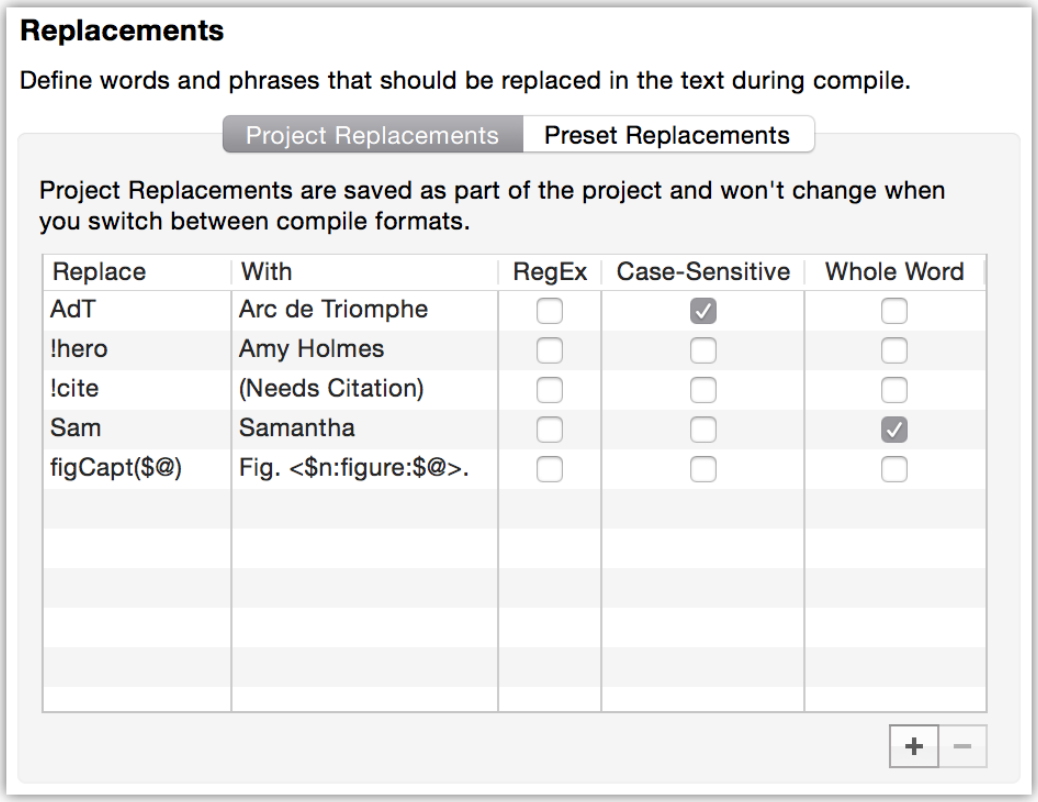

Figure 24.14: Replacements compile options pane

- *RegEx*: enables the regular expression engine (subsection 21.1.5) for both the *Replace* and *With* columns. When referring to stored values in the initial search pattern, use dollar-sign syntax, such as `$1`.
- *Case-Sensitive*: when checked, the letter case in the *Replace* column must precisely match, otherwise it will be ignored by the compiler.
- *Whole Word*: restricts the match to only those incidences where it has a clear space around the phrase. "Sam" will not match "Sammy" when this option is active.

Some example usages would be to replace an abbreviated version with a full proper name, to ease typing it in frequently, easier usage of placeholder tags or inserting common editing notes.

Two tabs are presented to you, "Project Replacements" and "Preset Replacements". It is important to know that while it may sound otherwise, the Replacements in the Preset tab are initially stored in your project as well. Everything in the compile pane is stored in

your project. The fundamental difference between these two tabs is what happens when you work with other presets. Preset Replacements will be saved into custom presets when you create your own or update an existing one, but Project Replacements never will. Likewise, Project Replacements are totally protected from any changes made to your compile preset. If you switch from the template settings to an outlining preset, for instance, the Preset Replacements will be reset (usually to blank), but the Project Replacements will be safe. If you create a Replacement in one tab, but later decide that it should be in the other, you can use normal copy and paste to exchange settings between tabs or even between projects. So in summary:

- *Project Replacements*: always stick even if you change presets, and are never saved into a preset. These are best for things that are highly specific to a particular work, like name substitutions or expansions specific to the project.
- *Preset Replacements*: are saved into your project file as well, but are volatile in that changing a preset will reset the list to whatever that preset defines (including nothing, if that be the case). These are best for things which suit an entire category of works.

An example of how Preset Replacements work:

1. You have a project you created using a built-in template. This template comes with a few replacements built-in, but you decided you want to add your own. At this point the project no longer using the preset, it has now "forked" into its own unique set of settings, and therefore the "Preset Replacements" tab is disconnected from the original preset
2. You remove one of the built-in replacements you never use, and add a new preset replacement you'd like to use in future projects
3. To make these changes available in the future (or to other projects you are currently working on), use the "Manage Compile Format Presets" option in the "Format As" drop-down at the top of the compile pane (section 24.5). Let's say you call this "Research Article". Now your project is once again synced with a format preset on the disk, and that preset will be available to other projects.

At this point you could make a new project and select that preset, acquiring all of the preset replacements available from it. If you ever wished to update it, however, you would not only need to add the items to the Preset Replacements tab, but *update* the preset on the disk as well. Any project using that preset will benefit from the new presets, but those

that *once* used it and no longer are (because you've customised the settings in one way or another) will need to be refreshed from the newly updated preset, or updated manually by copying and pasting Replacements between the compile panes of two open projects.

Replacements are evaluated in the order they are defined. This means you can use use earlier replacements in your subsequent replacements, if you are feeling brave.

> **Looking for what you cannot see?** If you need to include whitespace characters, such as carriage returns or tabs, in your search or replacement field, it is possible to do so by holding down the Option key and then pressing the Return or Tab key. You will not see the character itself, but it should impact the contents of the field in an expected fashion.

Replacements can contain other placeholder tags, as demonstrated (Figure 24.14) in the fifth row, which in that example will produce a numbered figure caption prefix, and also introduces an advanced method which will be detailed in the next section.

24.18.1 Advanced Replacements Usage

Replacements can also take a special tag, `$@` which will be used to match everything between the rest of what you type in the search field. What is found to the left and right of the `$@` tag is discarded as per normal, to be replaced by the text in the second column, but whatever was matched by the `$@` wildcard will be stored, and can be used in the second column to print what was stored in the text.

This is probably much more easily explained with an example. Let's say you have a replacement search for `^$@^`. Now in your base text, if the replacement field encounters a caret symbol, it will seek for a second caret symbol and then copy everything in between those two symbols into the `$@` tag. This can be placed into the "With" column, amidst any other characters you type in. A practical example that would be useful for those writing directly in HTML:

```
^$@^
```

Replaced with:

```
<span class="index">$@</span>
```

Applied to a little context from the text in the editor, we can take the following sample sentence and get the listed result upon compile:

```
This is a ^test^ of advanced replacements.
```

Replaced with:

```
This is a <span class="index">test</span> of advanced replacements.
```

The $@ character may be placed anywhere within the "With" field, but must be completely contained within delimiters in the "Replace" field. It can also be used multiple times within the "With" field. This means you can capture some text and then use it multiple times in the output. Here is a possible example, using LaTeX:

```
^$@^
```

Replaced with:

```
\index{$@}$@
```

Using the above example text, the output would be:

```
This is a \index{test}test of advanced replacements.
```

If you require more power when searching and replacing, you should consider using the Regular Expression syntax instead of the basic string matching engine.

24.19 Statistics

When using special placeholder tags which expand to display various statistics about the project, the options in this pane fine-tune how they work. Word and character counting tags can be inserted anywhere in your project, including some of the compile pane fields, like headers and footers. For a complete list of available codes, either view the help sheet for placeholder tags in the Help menu, or experiment with the various options available in the Edit ► Insert ► sub-menu.

Do not include front matter in word and character counts Enabled by default, this excludes any material in the selected front matter folder from statistics. Typically included would be items like the table of contents, acknowledgements, preface, and other material which is not generally counted as being a part of the main book. If for some reason you are using front matter for material which should be counted, then disable this option.

Word and character counts include all text This is the default behaviour. Any text that is set to be compiled as a part of your manuscript will be included in the

word/character count. This means that if you enable, for example, Notes or Synopsis export in the Formatting pane (section 24.11), the note text will be added to the global count as well.

Word and character counts include To adjust the scope of the counter, enable this option and then select from the following list of inclusive options. A checkmark next to the type of content means that it will be included in the total count. Counting filters will not impact what gets compiled, but only what out of that compile gets counted.

- Main text. To adjust counting for footnote, endnote, or comment text, see below.
- Notes
- Synopses
- Only documents matching criteria. This option functions in a manner identical to the global content filtering option (subsection 24.7.2). Consult the section on this feature for full documentation on how to use filters.

Count footnotes; Count comments and annotations Footnotes and endnotes will be included in the total count; this is the default. Likewise if enabled, comments and annotations will be included in the count.

Do not count spaces in character count Enable this option if you require strict character counts. If you are unsure, check with your publisher for which standard they use.

This option pane is a core feature available to all export formats.

24.20 Tables

These options are mainly useful for formats which require a single table for the whole document (such as certain documentary script formats), and should generally be left unticked. Use of this option allows you to use Scrivener's outline features freely, with numerous table based documents, and sew them all back together in the final compile.

Stitch together adjacent tables if possible If enabled, will cause any tables that are separated by only empty lines and blank spaces to be merged into a single table, provided that they have the same number of columns. For consistency, the merged table will use the column widths of the first table throughout.

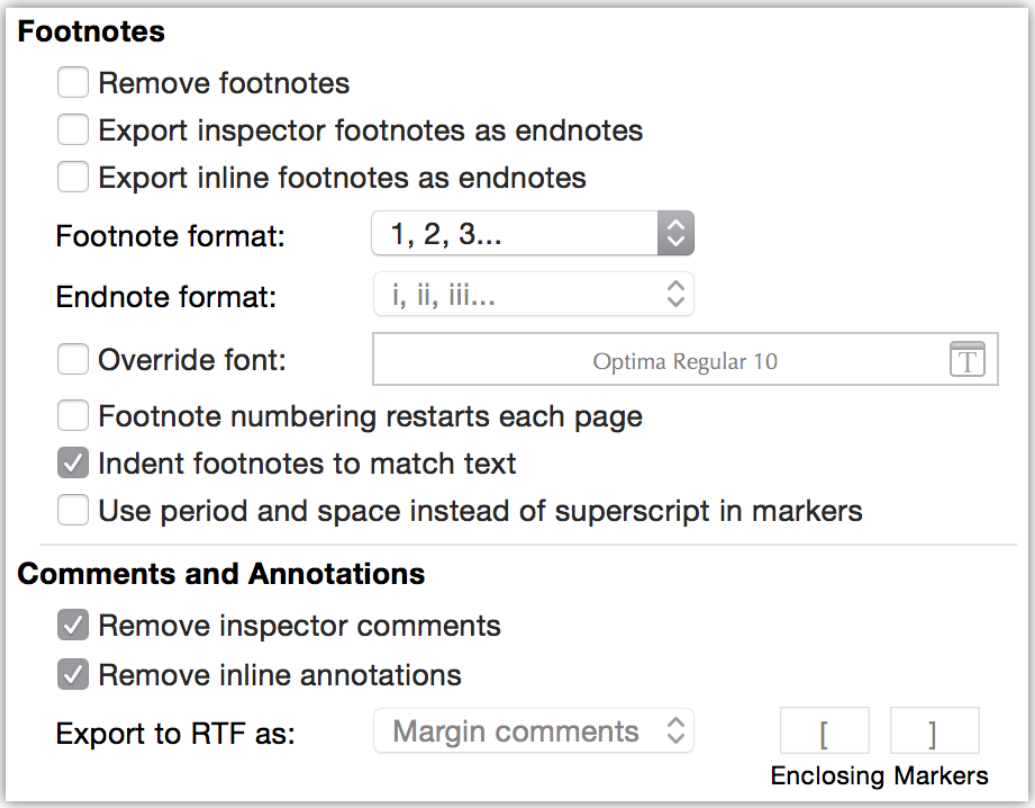

Figure 24.15: Compile—Footnotes/Comments Pane (RTF)

Insert blank row between stitched-together tables A table row will be inserted using the cell border characteristics around it.

Restrict width of stitched-together tables to page width Use this to cause the table to resize itself if you change the page size.

Convert tables to images with maximum width Used only for .epub and .mobi output, tables can be converted to static graphics for a more consistent display between the many e-reader devices available. You can specify the maximum width, in pixels, used to generate the graphic. The default value of 600 is considered adequate for most devices.

24.21 Footnotes/Comments

The Footnotes and Comments pane controls how these two forms of notation will be handled in the compile process, or whether they should even appear at all. Since there

are two ways to add notes to documents (linked or inline notes), it is with some formats possible to treat them differently. As with other areas of the application and documentation, the usage of the term "footnote" is equally applicable to "endnote", if the options to use endnotes have been applied.

The first section concerns itself with how footnotes should be arranged in the manuscript, and the second section deals with inline annotations and linked comments. Since not all of the available export formats support all available options, some variants of the pane will appear differently than shown (Figure 24.15).

This option pane is a core feature available to all export formats except for FDX, Fountain and FCF, where footnotes are never used.

24.21.1 Footnote Options

Some formats do not support real footnotes or endnotes. When using these formats, Scrivener will create the appearance of endnotes by embedding look-alike text representations for both the marker and the footnote itself. Since most of these formats also do not have a concept of a "page", it will be impossible for them to generate true footnotes. Consequently, they will nearly always be inserted at the very end of the document, becoming in effect, endnotes. When using these formats, the available options will reflect what is capable.

The HTML based formats, including e-books, will insert hyperlink references leading from the footnote marker to the footnote text. This will allow readers to click on the marker in the text and jump directly to the footnote. With e-books, most e-book readers have a "back" button which will let readers navigate back to the place there were before referencing the footnote.

Remove footnotes The exported manuscript will have all footnotes (inline or linked) stripped from it. When this option is enabled, the rest of the footnote section will be disabled.

Export inspector footnotes as endnotes When enabled, this will export all linked inspector footnotes as endnotes. If the next option is not set as well, this will require using a word process that can handle both endnotes and footnotes in the same document.

This option is only available to the RTF format, and by extension the DOC (but not DOCX) format, when RTF-based formatting is set in the Import & Export preference tab.

Export inline footnotes as endnotes Similar to above, but only exports inline footnotes as endnotes. If you only wish one type of notation to be used in your manuscript, make sure to check both of these options one way or the other.

This option is only available to the RTF format, and by extension the DOC (but not DOCX) format, when RTF-based formatting is set in the Import & Export preference tab.

Footnote format Using the drop-down menu, you can select how footnotes should be numbered in the final manuscript. For formats which support real footnotes, this will set an instruction in the manuscript file which word processors need to read and understand in order for the setting to have an effect. For formats which do not support footnotes (such as RTFD and TXT), this option will impact how Scrivener numbers items itself, as it creates plain-text footnotes.

Endnote format All of the notes above pertain to endnote numbering as well. If you document has both footnotes and endnotes, Scrivener will set two different numbering hints within the manuscript file; it is up to the word processor to handle them correctly.

This option is only available to the RTF format, and by extension the DOC (but not DOCX) format, when RTF-based formatting is set in the Import & Export preference tab.

Override font If you wish to use a separate font to print footnotes, this can be set here. Note that many word processors have their own special stylesheets for handling footnote appearance, so this option may not always be necessary. This format will be used consistently for all footnotes, even if they are featured in sections which have substantially different base formatting than the rest. Ordinarily, footnotes pick up the paragraph settings of the paragraphs they fall after.

Group footnotes... This feature allows you to gather footnotes together into a determined place, for formats that do not otherwise support true footnotes. It will toggle the default behaviour whereby all footnotes are attached to the very end of the manuscript, and will instead insert them at the nearest subsequent page break, becoming in most cases, chapter-based endnotes.

Available to Print, PDF, RTFD and TXT. The e-book formats have a similar option, but it behaves slightly differently and will be documented in the following section.

Add separator between text and footnotes Inserts a series of hyphens to set apart the endnotes from the main document, when using the plain-text format.

Footnote numbering restarts on each page Instead of keeping a running tally throughout the manuscript, each page's footnotes will restart according to the numbering methods available.

Indent footnotes to match text By default, footnotes will be first-line indented to the same degree as the directly preceding paragraph, according either to the formatting rules defined, or the inherent formatting in the document if the compiler is set to not reformat. When disabled, footnotes will be flush left at the margin, regardless of any other formatting settings.

Use period and space instead of superscript in markers This option conforms to the Chicago Style for note markers foot or endnote itself (not the main body text).

24.21.2 E-book Footnote Options

By and large, e-books use a similar scenario to those formats which cannot display end-of-page footnotes. Since e-books do not have literal pages they must use some form of endnotes, which will be cross-referenced with hyperlinks to and from the note and original text. The special options pertaining to these formats are:

Reduce marker font size The reference marker in the main text will be shrunken when this option is enabled.

Endnotes page title The section name for the table of contents entry which will contain all notes from the entire e-book. By default this uses the conventional "Notes".

Group endnotes by section with subheading Within this endnote section of the e-book, notes can be sub-divided into portions, using the name of the binder item they came from as the section header. Without this option, all endnotes will be displayed together in one long list.

24.21.3 Comments and Annotations Options

As with footnotes, it is possible to use both inline annotations and linked comments in the same project, and selectively handle them during compile. You may wish, for example, to use inline annotations for simple editing notes, and linked comments for references to source material. It might then be useful for collaborators to see the source material references, but not the simple inline editing comments to yourself.

Most of the compile formats have the same options available, but will handle the export of annotations and comments in different ways, depending on what is supported within that format.

- *RTF/DOC*: Provide additional options for utilising margin comments. Unfortunately this feature is not widely supported by other programs. If comments refuse to show up in your preferred word processor, try using one of the other methods listed below.
- *Other rich text and word processor formats*: In all of the formats that support the ability to set font colours, annotations and comments will be inserted as actual text in the document, enclosed in markers, and then coloured according to the annotation text colour, or linked comment background colour, in the project. For the purposes of this definition, HTML, WebArchive, and the e-book formats are considered "rich text".
- *Text based formats*: For formats like plain-text and MultiMarkdown (when the HTML method is disabled), annotations and comments will be simply placed into the text where they originally appeared in the project. You can control what, if anything, is used to enclose the comment in the text (square brackets are used by default).

The following options are available:

Remove inspector comments Only inspector comments will be stripped from the resulting manuscript.

Remove inline annotations Only inline annotations will be stripped from the text. If you wish for all comments to be removed from the manuscript, make sure both this and the above options are checked.

Export to RTF as Several options are available for handling all annotations and comments. This drop-down menu is only available to the core word processor formats.

- *Margin comments*: When using a word processor that can read margin comments, it is most often a good idea to use this option, as comments will be placed out of the way and not be regarded as part of the actual text. Some programs will even provide handy comment navigation, to make review easier.
- *Inline comments*: If comments are not showing up for you using the above option, this method will produce comments using the method described above for other rich text formats. This method is the safest to use, but will embed the content of comments directly into the manuscript text, and is thus only useful for proofing scenarios and scattered other workflows.

- *Footnotes*: All annotations and comments will be inserted into the footnote stream along with any other footnotes (if any exist). This option shouldn't be used unless it doesn't matter if commentary and footnotes commingle. For purposes of ordering, they will be inserted into the stream relative to the position of the footnotes/endnotes around them.
- *Endnotes*: This works the same as with footnotes, only they will be inserted into the endnote stream.

Inline annotation enclosing markers If annotations are to be left in place, you can set the two markers that will be used to delineate them from the rest of the text. Someone who is publishing in HTML might for instance wish to use the open and close syntax for HTML comments. Those using MMD to create LaTeX files might do the same, as HTML comments allow one to write raw LaTeX into the editor. This option will be designated as "Enclosing markers" when using a word processor format and "Inline comments" selected, above.

24.21.4 MultiMarkdown Options

The MultiMarkdown compile formats handle footnotes uniquely from the other formatters. Scrivener will, by default, automatically generate and ID the necessary footnote syntax for you. An MMD marker will be inserted wherever the footnote has been placed in the document, and the appropriately matching footnote text syntax will be inserted elsewhere in the compiled MMD file.

Whenever Scrivener is set up to work in this fashion, linked inspector footnotes will use a different ID prefix than inline footnotes. In ordinary MMD usage, this will make no visual difference to you. MMD does not care what the footnote ID is, so long as it is unique to the document. However, using a distinct prefix for linked footnotes gives you the option to write an advanced XSLT file to handle them differently (such as mixing endnotes with footnotes), if you wish to.

Inline footnotes will be prefixed with "fn", followed by a serial number. Linked footnotes will be prefixed with "cf", again followed by a serial number.

Convert to HTML-styled text Since MultiMarkdown has no good analogue syntax for comments, by default Scrivener will insert a styled HTML container around the comment. Both inline annotations and linked comments will be treated identically in this fashion. Since MMD will pass HTML straight through its engine, the result will be a range of text coloured with whatever colour the annotation text or comment background was set to. The appearance, sans the bubble around the text, will thus be identical.

When this option is disabled, annotations and linked comments will be exported in the same fashion they are for plain-text files; enclosed in the defined markers. This latter option should be used with MultiMarkdown version 3, which is no longer XHTML based when using non-HTML target formats. Using HTML styled inline annotations in MMD3 will cause them to disappear from the output. Use the enclosing marker fields to instead inject code of your choosing. You could for instance use an open and close marker pair of: <!- \marginpar{ and '} ->, to convert your inline annotations to LaTeX margin notes.

> **Pro Tip**: In cases where an annotation falls entirely on its own paragraph, Scrivener will use a styled div instead of a styled span elements to wrap the comment in. This means you can use complex MultiMarkdown within annotations that are on their own. Annotations embedded within a paragraph of otherwise normal text, in any way, will use spans in order to preserve the original document flow, and thus cannot contain complex syntax. They can however utilise inline formatting such as bold, italic, footnotes, and so forth.

24.22 Page Settings

Various print and page settings can be set up with the Page Settings pane, such as margins, paper size, headers and footers. Only formats that support page settings can use this pane, and amongst those not all support headers and footers. For these, that part of the pane will be removed.

Use project Page Setup settings The project's Page Setup settings (File ► Page Setup...) will be used to determine the paper size, printable area, and margins. Disabling this option will enable the margin and compile page setup buttons below.

Page Setup button Here you can set the physical size of the page and other basic printer settings, just as in the standard Page Setup... feature. Refer to Printing (chapter 26) for further documentation on how to use these features.

The Page Settings button will not be available for the FDX format, as it does not store paper size in the document.

Margins Margins set how far from the edge of the paper stock the text container will extend to. This measurement does not factor in the non-printable area, which is printer specific, so when using very low values it is a good idea to check how close to the edge of the paper your printer can go, to avoid truncating the text.

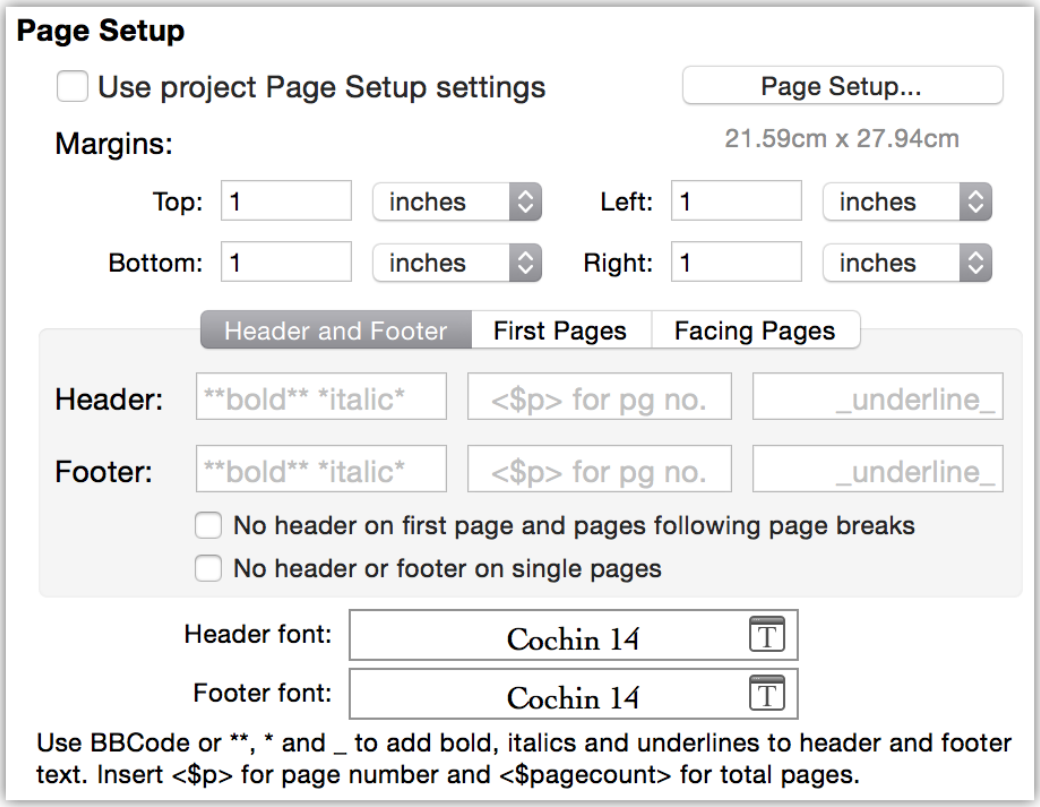
Page Setup
Use project Page Setup settings
Page Setup...
Margins:
21.59cm x 27.94cm
Top: 1 inches
Left: 1 inches
Bottom: 1 inches
Right: 1 inches
Header and Footer
First Pages
Facing Pages
Header: **bold** *italic* <$p> for pg no. _underline_
Footer: **bold** *italic* <$p> for pg no. _underline_
No header on first page and pages following page breaks
No header or footer on single pages
Header font: Cochin 14
Footer font: Cochin 14
Use BBCode or **, * and _ to add bold, italics and underlines to header and footer text. Insert <$p> for page number and <$pagecount> for total pages.

Figure 24.16: Compile—Page Settings (RTF)

When using the Facing Pages tab (subsection 24.22.4), symmetrical margins can be used. The right and left values will be used in alternation as is suitable for many printing specifications.

The following controls cannot be accessed for RTFD. If the Aspose converters are disabled, only RTF will make use of them.

24.22.1 Header and Footer Tab

The header and footer area is divided into three tabs, the first tab "Header and Footer" will control these settings for the entire document. The second tab, "First Page" will control how the first page is considered for page counting and secondary header/footer settings. Finally the third tab, "Facing Pages", can enable recto/verso layout controls for symmetrical header/footer control and margin offset alternation.

Each of the tabs are very similar to one another: There are two rows for the header and footer, each with three fields which allow you to enter different information at three

different points of alignment: left, centre, and right, respectively. It is possible to have all three in a row in use at once.

Formatting can be done in these fields using markup. You can choose between using simple BBCode, or Markdown, to indicate bold, underscore and italic ranges. Examples have been provided in the placeholder text for these fields. You can also use tokens in the header and footer fields, some of which are only available to these fields.

> **Using Special Characters**: Since markup is allowed in these fields, some special punctuation marks set aside for markup cannot ordinarily be used. You can however instruct the compiler to ignore special characters. If you need to print asterisks or underscores, you can wrap the fields in double-curly-braces. The entire row needs to be treated this way. So for example if you wished to turn off markup for the header, you would type in "{{" in the beginning of the left-aligned field, and "}}" at the end of the right-aligned (third cell) field. This would need to be repeated for the footer if desired.

If the following options are available to PDF, Print, RTF and the word processor formats using the improved converters, but not RTFD.

No header on new pages following page breaks When enabled, the header (not footer) will be disabled on any page following a page break. Most often this will be used to keep the title page for chapters clean, as is common in typesetting.

No header or footer on single pages When enabled both the header and footer will be removed from any page that has a page break directly following it. This would most often be seen in cases like book or part level breaks, where a single page is dedicated to some sort of title.

Header and footer fonts Customise the font used for the header and footer. With RTF and the derived word-processor formats, this font must appear at least once in the document elsewhere. This requirement is not necessary for PDF and Print.

The tokens (Table 24.5) listed as available only to headers and footers only, have an alternate usage whereby if the token name is typed in using all-caps, the final result will be transformed to all-caps as well. For example, if the title of the Draft folder is "My Novel", and the token <$COMPILEGROUP> is used in the header, it will be printed as, "MY NOVEL" in the manuscript.

24.22.2 Sectional Page Headers

When using the PDF compile format, the <$sectiontitle> placeholder tag in the header and footer fields will print out the Binder name (including any suffix or prefix) of the

Table 24.5: Useful Header and Footer Tokens

Token	Description
	Available to Header and Footer fields
<$compilegroup>	The current compile group. If you wish for this to be more descriptive than "Draft", you can change the name of the Draft in the Binder to be the name of your book.
<$projecttitle>	Project Title, as set in **Project ▶ Meta-Data Settings...**, or falls back to the name of the project file itself if none has been specified.
<$abbr_title>	Optionally defined abbreviated project title, also defined in the project Meta-Data settings panel. This also falls back to using the project title above, or the filename of the .scriv project, if necessary.
<$sectiontitle>	When compiling to PDF, this will print the title of the last Binder item that used a page break. All subsequent pages will continue printing that title until a new page break is encountered. When used with RTF, this feature does not work, so the token will print using <$abbr_projecttitle> instead.
<$surname>	The surname from the current user's Contacts entry, or the project's meta-data settings.
<$forename>	The forename; as above.
<$fullname>	The full name; as above.
	Some useful global replacement tokens
<$p>	Prints the current page number
<$pagecount>	The total page count for the entire manuscript. This is a static number that is primarily useful in conjunction with the page number token. A value of "<$p> - <$pagecount>", will produce, "73 - 258" on page 73 of a 258 page manuscript.
<$p>	Prints the current page number using Roman numerals. You can alternatively uppercase this placeholder when typing it out, to produce uppercase Roman numerals.
<$shortdate>	The current date, according to the system short date settings.
<$mediumdate>	As above, using the system medium date settings.
<$longdate>	As above, using the system medium date settings.
<$wc>	All of the word and character count tags can be used in the header or footer.

item which last caused a page break. In practice, this will often mean the folder in which a section is contained, but anything that causes a page break, whether procedurally via the Separators pane (section 24.9), or by hand with the "Page Break Before" inspector checkbox, will reset the value of this placeholder to its title. All subsequent pages will use this title until another page break is encountered.

This feature, due to technical limitations, is only available to the PDF compile format. When used with the other export formats, the value of this placeholder will fallback to <$abbr_projecttitle>, which itself will fallback to the project title if necessary. In this fashion, you can safely use the section title placeholder in your headers, even if you intend to produce both PDF and RTF compilations. When compiling with PDF you'll get dynamic section headers, and with RTF you'll get the abbreviated project title, or the full project title itself if an abbreviation has not been set up in the Project Properties (section 10.4) panel.

24.22.3 First Page Tab

This tab contains an identical set of fields for header and footer, which must be enabled with the "Different first page header/footer" option. Each field here can specifically override the settings in the prior tab on a field-by-field basis.

Page numbers count first page This option will cause the page counter token to start counting at the first page, rather than skipping it, even if the first page does not display a page number. If you are not including a title page in your manuscript, this option may be useful.

Start regular header and footer on... Only available to the PDF and Print formats, this option lets you offset the alternate header footer settings by additional pages. The "Automatic" selection is the default, and in most cases it is safe to leave this alone if you use Scrivener's other book management features, such as front matter (subsection 24.7.3) selection. When front matter is in use, the above header and footer settings will be used for the entirety of it, with the normal header and footer (defined in the first tab) being used for the remainder. This way you can add or remove material from the front matter, or easily switch sets, without having to fiddle with this setting and count pages.

When you are not using front matter, this setting will be equivalent to '2'.

24.22.4 Facing Pages Tab

When "Use facing pages (alternating margins, headers and footers)" is enabled, the header and footer grid will describe the settings for the *verso* page, or the left-hand even numbered page in a printed book. The initial header and footer grid in the first tab will

become the *recto* page, or right-hand odd numbered side. This will make it possible to put a page number in the *verso* top-left corner and *recto* top-right.

Additionally, when this option is turned on, the margin settings defined above will alternate. If "Left" margin is set to 0.75" and Right to 1.25", then on the *verso* page the right margin will actually be 0.75" and the left 1.25". This alternation will not be directly displayed in the settings above; you will just need to visualise the result.

24.23 Compatibility

This pane is used for setting a few advanced MultiMarkdown options. The available options are listed below, along with the compile target format they will appear with.

Use MultiMarkdown 2 Compatibility Only available when the basic "MultiMarkdown" compile for target is chosen, as MMD2 can no longer be initiated from the front-end. This will cause the compiler to generate MMD2 compatible syntax. You will need to use the command-line tools to post-process the product file.

Use XSLT Post-Processing *Available for LaTeX, HTML and PDF.* If enabled, Scrivener will use alternate commands which will first generate an HTML document, and then pass it through the XSLT processing engine using the XSLT template you have provided in the meta-data block. This option will function without the MultiMarkdown support package installed in your Application Support folder, but if you do have one, Scrivener will use your copy instead of its built-in copy, allowing you to easily modify the scripting flow and XSLT files.

Markdown Compatibility Mode *Available to HTML.* Causes the MMD engine to produce XHTML output similar to default Markdown, rather than the updated HTML5 output that MultiMarkdown now produces by default. This will mainly be useful if you have CSS stylesheets or post-processing already set up in assumption of Markdown standard output.

Show PDF Log File *Available for PDF.* Writes the output of pdflatex's output to a file. This will be useful if you are running into typesetting errors and cannot get Scrivener to generate a PDF out of the box. Note that in many cases you will have to fix certain problems in the output .tex file, and generate a PDF using another tool such as TeXShop.

24.24 Meta-Data

Meta-data is used to add properties to your manuscript, which many applications will display in a document properties viewer, and with some formats, will be used to add

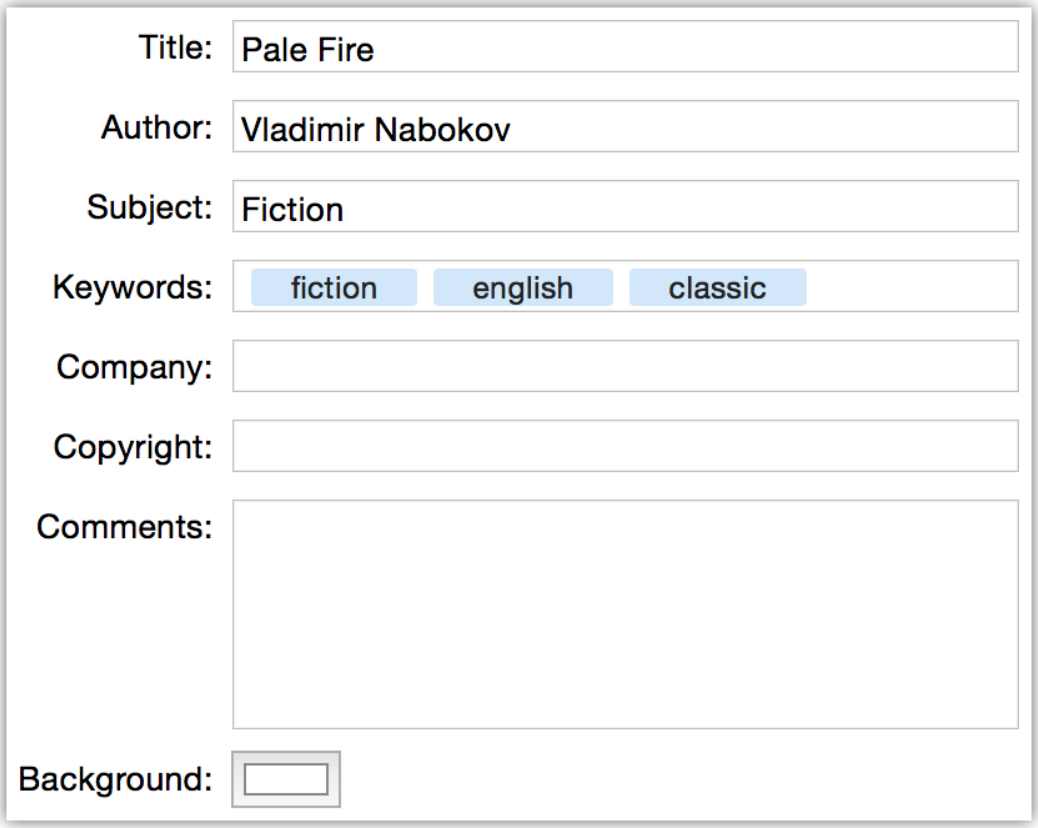

Figure 24.17: Compile—Meta-Data (General)

searchable criteria to the Spotlight database in the document's properties. There are three primary meta-data interfaces available to different formats. The general meta-data interface will appear for all formats which support the standard set of property fields, such as RTF & RTFD, the word processor formats, and HTML & WebArchive. The other two meta-data panes will only appear along with the specific formats they have been tailored for, MultiMarkdown and ePub/Mobi. Despite the differences, the various panes are referred to by the same name in the option pane list.

Since, with the exception of MultiMarkdown, meta-data is not meant to appear in the content of the manuscript itself, none of the plain-text formats, or any formats that do not support auxiliary properties, will show the Meta-Data pane.

24.24.1 General

With the exception of the Keywords field, the provided fields are all free-form, though keep in mind that in some cases, like Spotlight searching, the content for the field should

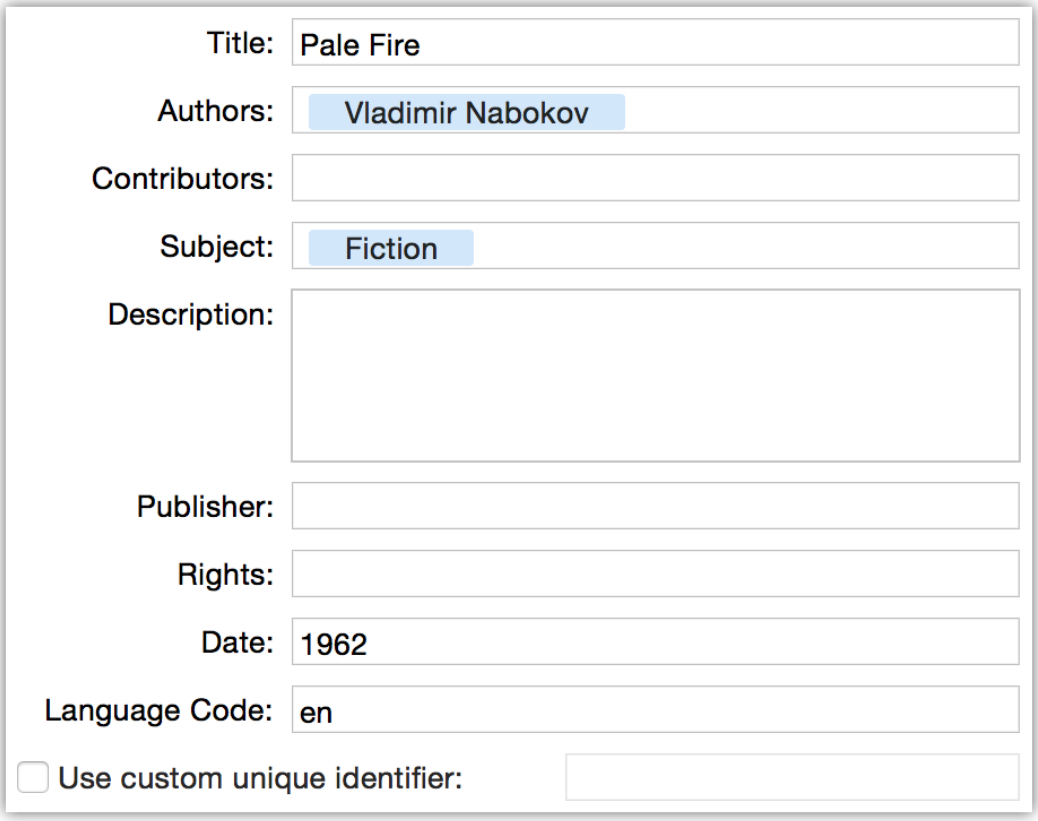

Figure 24.18: Compile—Meta-Data (ePub)

match its intended purpose in order to keep indexing catalogues logical. The Keywords field takes values separated by commas, which will be shown as blue "pills" in the interface as you enter them.

The Background field will let you set a paper colour for the manuscript. This is, by and large, ignored by most word processors, and is primarily useful for the HTML and WebArchive formats.

24.24.2 E-books

Most of the fields in the e-book meta-data pane are free-form text fields. They will be inserted into the appropriate book description fields, which will be available to e-readers to display in whatever manner they provide, and used to automatically sort the book within electronic catalogues. The Title and Author fields are required, and will be supplied with default values (taken from the **Project ► Meta-Data Settings...** panel) if the text fields are left tempty in this pane.

The three fields which you will want to be more careful with are Language Code, the custom unique identifier, and if you have multiple contributors, the Authors and Contributors fields. In the Language Code field, make sure you use the correct ISO two-letter abbreviation for the language the book is written in. If you do not wish to provide a custom identifier, Scrivener will generate one for you based on the author, title, and date. If your publisher has provided a unique identifier for you, then you can check the box and type in your own.

For multiple authors and contributors, separate each author with a semi-colon to ensure that they all get registered as separate individuals in the e-book. The ePub format allows for multiple authors so that book searches can be done on each author independently. Amazon's KindleGen utility does not recognise multiple authors, so only the first name contained within a bubble will be applied to the .mobi book. To ensure all authors are listed in the .mobi file, refrain from using semi-colons to separate the names, or add them later with a tool that is capable of adding multiple names to the .mobi file you've compiled.

24.24.3 MMD

The Meta-Data pane for MultiMarkdown compilations provides a front-end to the preliminary meta-data block at the top of most MMD files. A few examples are inserted into new projects by default. Note that what you type into the text area below the field will be printed into the compiled document verbatim. Thus, all rules pertaining to the meta-data syntax should be followed. Empty lines in this area will cause the meta-data block to prematurely abort.

Individual rows can be moved around with click and drag. The order of meta-data fields only matters when using MultiMarkdown version 3.

You may also paste raw text meta-data into this table and Scrivener will parse it and turn it into keys and values for you. For example, if you copied in a meta-data block containing:

```
Title: The History of Origami
Author: Ioa Petra'ka
Date: March, 2011
Revision: 0.5-proof
```

Then you would see a result similar to the one depicted in the screenshot (Figure 24.19)

This interface, being project-based, is useful for creating a general meta-data block. If your project is actually comprised of multiple documents, or if you simply prefer to type in the meta-data block by hand, you can use documents titled "Meta-Data" in the binder.

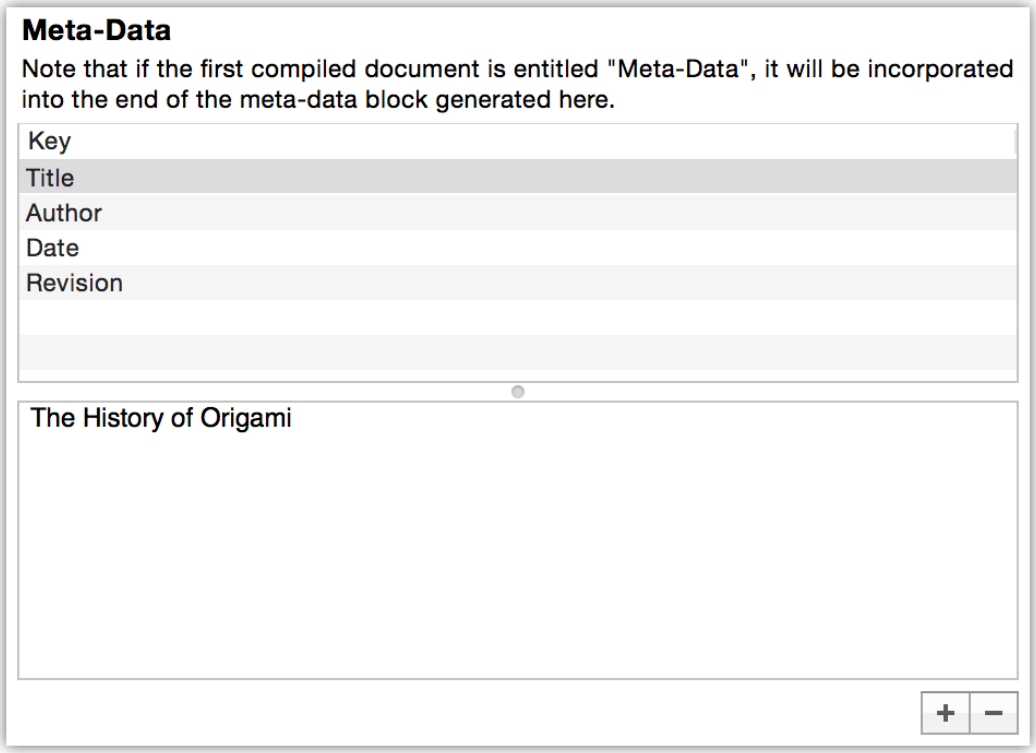

Figure 24.19: Compile—Meta-Data (MMD)

Whenever a document so titled is the first document in the compile group, it will be appended to the meta-data provided in this pane. Duplication should be avoided where possible. A good strategy for multi-file projects is to use the project compile settings for general meta-data which does not change (such as your name), and then the keys specific to each output file in the "Meta-Data" documents.

For further information on the available MultiMarkdown meta-data fields, refer to the online documentation[10].

24.24.4 Fountain Title Page

The meta-data for this section is taken from the MMD meta-data interface.

24.25 RTF Compatibility

Scrivener's default RTF exporter supports features that some other products and free editors do not recognise. In a worst case scenario, this can result in files which do not

[10] http://fletcherpenney.net/multimarkdown/users_guide/multimarkdown_syntax_guide/

correctly load at all, display only a part of the content, or at the least omit the parts they do not understand. The following options can fine-tune the RTF file you create, so to better increase its compatibility at the expense of formatting. This option pane is only naturally available with the RTF format, and also to the .doc format when it has been set to use a RTF-based code, in the Import & Export preference tab (section B.10).

Strip table formatting from text Use this feature when tables are causing the RTF file to render incorrectly or not at all in the target word processor

The contents of the tables themselves will not be removed, but the table cells themselves will be. This results in a block of text that "flattens" the table contents into a long list. Generally you will not want to use this unless the target application completely fails to render tables.

Flatten footnotes and comments into regular text Use this feature when the target word processor fails to properly display footnotes and/or comments.

When enabled, all footnotes and comments will be converted to normal text, rather than using RTF codes to insert them. Since the notion of a footnote requires pagination to place the footnote at the bottom of the page, the end result is that all footnotes will be exported as endnotes. Reference markers will be inserted into the text using standard punctuation to do so. This feature modifies the existing behaviour of your compile settings. If you have opted to strip out all comments, checking this box will not override that, they will remain excluded. It only modifies how the feature is exported, if it is scheduled to do so.

Use Word-compatible indents for bullets and numbered lists Use this option when working with Microsoft Word.

Word uses a different mechanism for displaying indents in enumeration and bullet style lists. This option will attempt to preserve as closely as possible the look and feel of your original document. If you are not using Word, and you are getting erratically formatted lists, try disabling this option.

Ensure hyperlinks are colored and underlined Most word processors will do this for you, but Microsoft Word will not, resulting in links that cannot be seen. Check this box to make hyperlinks visible in Word.

24.26 Quick Font Override

Available to all formats except those which do not use fonts (plain-text and MultiMarkdown), this option is coupled with the summary mode font selection which provides for

an easy way to override fonts throughout the entire compile setup. When in use, this will set every font family in the Formatting pane, header and footer settings, and footnotes—it will even override the font in documents that have been set to compile "as-is". A common example usage would be the adjusting the submission format preset to use Times New Roman instead of its default, Courier.

This option will not actually change the font selections that it overrides, so it can safely be used in a temporary capacity. Once disabled, all font declarations will go back to how they were set up initially. It also does not make any change to the font size or variant. Thus you will still need to use the font controls throughout the compile interface if you need to adjust sizes and variants (such as italics).

24.27 KindleGen

Available only to the Kindle e-book export format, this option pane is where you will inform Scrivener of where you have installed the KindleGen utility. Without this utility, it is impossible to create a Kindle e-book straight out of Scrivener, and you will not be allowed to make any changes to the compiler until you have selected the KindleGen tool. Amazon's tool requires an Intel based Macintosh computer, and will not run on PowerPC based computers.

To acquire this tool, download the software for your computer from Amazon's web site[11]. KindleGen is only compatible with Mac OS X 10.5 (Leopard) and greater, running on Intel hardware. It will be distributed in a ZIP file, which will you need to decompress (if you browser does not do so for you) and place in a safe place. It is a good idea to put it in your Applications folder so you can find it in the future if need be. Once the KindleGen folder has been installed on your computer, click the **Choose...** button to select the "kindlegen" executable from within that folder.

Save the KindleGen log file with the Kindle file If you are having difficulties getting the KindleGen program to produce a valid .mobi file, enabling this option can help you determine what is going wrong. The log file will be placed in the compile location you specify once you click the **Compile** button.

Scrivener does not verify if you have chosen the correct application, so double-check your selection as the wrong choice will cause errors to occur. A common symptom of having picked the wrong item is if the compiler generates an error about no .mobi file being present, after compiling, and switching on log reporting does not end up producing a log file, since KindleGen produces this log, not Scrivener.

[11] http://www.amazon.com/kindlepublishing

Save the source files in a folder with the exported Kindle file This is an advanced feature intended for those who wish to customise the look and feel or structural layout of their e-book after compiling it. The actual files used to create the e-book will be output as plain folders and files for you. These can then be edited freely, and then turned into a .mobi file using KindleGen yourself.

Chapter 25

Exporting

You can export your work and research material from Scrivener at any time, either as individual documents or by combining the draft into one long manuscript and exporting it in the format of your choice, using compile. Since the latter is a large topic in and of itself, it is covered in depth in its own chapter, Compiling the Draft (chapter 24). Here we will look at various ways to export data piecemeal from your project, either for backup purposes, or to facilitate collaboration with authors who do not use Scrivener.

25.1 Drag and Drop Between Projects

While it is not possible to drag items directly from the binder to other programs on your computer (including the Desktop), you can easily drag items between Scrivener projects. This will retain almost all of the document's meta-data, including its notes, label, status, snapshots, and keywords. If necessary, set up the target project to have the same custom meta-data settings as the source project. For example, if you intend to drag a file from project A over to project B, and this file has been assigned to a custom meta-data value of "Oslo" in a custom "Location" column, you will need to ensure that the "Location" column has been set up in project B's meta-data configuration prior to dragging. Scrivener will attempt to match labels and status first, and if it cannot find a like-named meta-data name, it will create one for you, retaining the original colour assigned for labels. If the label in the target project is named the same, but has a different colour, then the imported items will match the new label and their colours will change to match.

25.2 Exporting Binder Files

To export files and folders from the binder as individual files and folders on your system, select the files you wish to export in the binder and then choose **File ► Export ► Files...**, or press ⇧⌘–**E**. This will by default also export any descendants of the selected items as well. You can change this behaviour by using the "Export selected files only (do not export subdocuments)" option, which you'll find in the Export panel itself.

When exporting more than one file, Scrivener creates a folder on the disk to hold all of the exported files. Enter the name for this folder in the "Save As" text field, and then choose where you would like the folder to be created. From the "Export text files as" drop-down menu, you can choose to export text documents as one of the following:

- Rich Text with Attachments (RTFD)
- Rich Text (RTF)
- Microsoft Word (DOC or DOCX)
- OpenOffice.org (ODT)
- Web Page (HTML)
- Final Draft (FDX)
- Final Draft 5–7 File Converter Format (FCF)
- Fountain (.fountain)
- Plain Text (TXT)
- HTML

All media files will be exported as they are. You can also choose to export comments and annotations, meta-data, and/or snapshots; and optionally choose to strip out all notation. Notes will be exported as separate files using the text file format you chose above (RTF will be used instead, when any script formats are chosen), and the meta-data (including the synopsis, label, status and so on) will be exported into separate plain text (TXT) files. The structure of folders created on disk will reflect their structure in the binder[1], so in this way you can export all of your files from Scrivener for use in another application if you so wish, or as an archival backup.

[1]There are a few caveats to watch for. Since traditional files and folders do not support text material in a folder, Scrivener will need to create a separate text file to hold whatever material had been typed into the folder in Scrivener, if applicable. Another caveat is that items will be sorted (typically alphabetically) rather than using the original order in the binder. If retaining the original order is important, producing a compiled version of the document alongside the full export would be a good idea.

25.3 Exporting to an Outliner with OPML

Many outliners support a common format known as OPML. In a sense it is like RTF in that it is a way for many different programs to communicate with one another, and in this case specifically to communicate outlines. It can describe "headings" which are what you see in the Binder as names of items, and the relationship between those headings in terms of order and depth. There is an additional convention on top of this format which allows for some outlining programs to attach "notes" to outliner headings. Unfortunately this method of attaching notes is not a standard, and so support for it may vary in terms of quality and features. The good news is that many applications these days do support a "_notes" field.

Scrivener supports the "_notes" method for attaching additional data to outliner headers, both on import and export. This is optional, so if you are having difficulties loading the OPML file in the target software, try using "Titles only" as your export option. This mode will adhere to the standard OPML specifications.

Typical usage is to select (binder sidebar, corkboard, and outliner views are all acceptable places to make a selection) a container that you wish to export (all descendant items will be included), and use the **File ► Export ► OPML File...** menu command. You may also indicate that the entire Binder should be exported. Outliners use a "root" item concept. When you select a container to be exported, that container will be considered the "root" item. Depending on what outlining software you use, this item may not be initially visible, as some default to hiding the root item. It is however stored safely in the OPML file. When exporting the entire Binder, Scrivener will create a new root item to represent what was the Binder itself, and place all of the top-level items within it (even the Trash and its trashed items).

When you indicate that some extra form of textual data should be added to the outline, beyond the title headings themselves, they will be attached as notes. Because of limitations in the design of this convention, it is not possible to transfer rich text information between outliners using the "_notes" method. All material will be converted to plain-text.

25.3.1 Usage

The following options are available from within the export panel:

Export entire binder Rather than use the current selection, export the entire Binder outliner from top to bottom. This will include *everything*; even items you have currently in the Trash can.

Note attachment options Using the loose convention of placing plain-text notes into a "_notes" attribute for each outline heading, you can often communicate extra

information to other outlining software. In all cases, the associated data for that option will be assigned to the same binder item it came from. So if an item named "Fruit" has a description written into its index card, and you select "Include Synopses", then in your outliner, the "Fruit" heading will contain a note field with the description from the synopsis attached.

- *Titles Only*: if attempting this gives you issues or error messages, the first option "Titles only" conforms most closely to the published specifications for OPML and should be the safest format to use.
- *Title and Synopses*: contents of any index cards, or the synopsis field.
- *Title and Text*: in this case, "text" refers to the main body text for that item, or what you have typed into it using the text editor. This will be the content of the research material, or portions of your draft as you have written them. This only includes *text* from file or folder items in the Binder.
- *Title and Notes*: uses the auxiliary Inspector Notes field for each included item. As with the main text option, the results of this will be converted to plain-text.

25.3.2 Limitations

- This method is intended for exporting an indented outline, rather than exporting meta-data. If you wish to export tabular meta-data lists, try Exporting Meta-Data to a Spreadsheet (section 25.4).
- Only text items (including folders, naturally) in the binder can include their content in the "_notes" field. Research items such as graphics or PDF files can export only their titles, synopses and document notes.

25.4 Exporting Meta-Data to a Spreadsheet

Outliner views can be exported to formats which are suitable for loading in spreadsheet software, like OpenOffice Spreadsheet or Microsoft Excel; many databases; and other miscellaneous programs that support compartmentalised data. If you wish to export a specific set of columns, or even all possible meta-data, use **File ► Export ► Outliner Contents as CSV...**. This menu item will be disabled unless an Outliner is currently the active view. This method is primarily intended for tabular data export. If you would prefer a hierarchal export more suitable for outlining, you might want to try Exporting to an Outliner with OPML (section 25.3).

Formats The three formats below are all commonly accepted by most spreadsheet applications. Check with your preferred software for details. Scrivener will export the first row in the file as "headers", so your importing software should be instructed as such if applicable.

- Comma-separated values (best for most spreadsheet software)
- Tab separated values (use this for sharing data with Tinderbox[2]).
- Semi-colon separated

Only include columns visible in outliner When checked, the column list and the order of those columns will be used to create the data file. When deselected, Scrivener will export *all* meta-data columns (including any custom columns you might have added) in the order that they appear in the View ► Outliner Columns ► sub-menu.

[2] http://www.eastgate.com/Tinderbox/index.html

Chapter 26

Printing

Given that there are two discrete functions in the binder: your draft or works in progress, and all of the support files and notes around it; you need two fundamentally different print modes. Since proof printing and final printing the draft are closely related to the act of compilation itself, this function has been built into the compile interface. For all other forms of printing, either piecemeal or in groups, the standard print mechanisms familiar to most applications will be available, including a few special-purpose printing tools for visualising clusters of data, like index cards and outliner columns, that specialise in highlight meta-data and synopses.

In most cases (compile, notably excluded) all you need to do to print a selection is press ⌘–P, or select File ► Print... from the menu. Continue reading for tips and features on how to control the appearance of your print-outs. Many aspects of the printout can be adjusted in File ► Page Setup..., and will be covered in each relevant section. These settings are saved into each project, not globally.

26.1 Document Printing

Individual documents and supported media can be printed one by one. When more than one text document has been selected in the binder they will be printed together according to the current view mode. If you are viewing the selection as a corkboard, index card printing will be used, if viewing them in scrivenings, the text view will be printed using the text document printing settings (below).

If you select a single media file, such as an image or a PDF, it will printed as per normal (as if you had loaded that file in its normal viewer and opted to print it. However if you select more than one media file, or a mix of media and text files, index card printing mode will be used instead, which you can read about in the following section.

26.1.1 Text Documents Print Settings

Page margins can be set up with File ► Page Setup... or ⇧⌘–P, and then select “Scrivener” from the “Settings:” drop-down menu.

To set up how documents are printed individually (or together, when printing from a scrivenings session), navigate to the above setting pane, and then click the Options... button in the lower-right. Adjustments for text document printing can be found under the "Text Documents" tab.

As you adjust these options, the preview pane above will adjust, giving you a preview of the final look.

Print Options These impact the overall look and feel of the document, including the headers, footers, and base font.

- *Print page numbers*: Insert a page number in the top-right corner of each page in the header area. This number will be relative to the selection you have chosen to print, it will not reflect the actual page number of the complete draft (if you are printing from within the draft).
- *Print the file name/date*: Inserts the selected document's name next to the page number in the header area. If more than one document has been selected, this will revert to "printing".
- *Remove annotations*: By default, annotations will be left in, using their original colour, and enclosed in square brackets to help identify them when using a black and white printer. use this option to strip them out of the printout.
- *Insert linked comments*: By default, linked comments will be inserted beside the highlighted anchor range in a darker variation of the highlight's colour. Uncheck this to keep them out of the print job.
- *Print using font*: If you wish to override the formatting of the documents temporarily, and use a uniform font, check this box and then select the font and font size in the activated option menus, below.

Elements to print Similar to the Formatting pane in the compiler, you can optionally add meta-data to the printout for each document that is selected. By default, only Text will be selected, which is the main text body of each document.

> **Printing with Placeholders and Number Tokens**: The basic one-off print tool will not evaluate replacement tags and auto-number counters in your documents. If you wish to print with these tokens, you should select the documents you wish to print, and then use Compile with either the content selection set to "Current Selection", or the use of "Current Selection" as a filter. Read about doing this with the compiler contents pane (section 24.7).

26.1.2 Other Print Settings

Since scrivenings mode is not available to media, your only options for printing groups of them will be on index cards or using the outliner printer. Consult the following sections for details on these two methods.

Most research files will print out in a standard fashion according to their type, and there are no options for adjusting how that occurs. However, images and web pages have a few options available in the same Options palette described above, under the "Other" tab.

Web Pages By default, background images and colours will not be sent to the printer to save ink and increase clarity. However if the background image is an integral part of the design, you can change this behaviour with this checkbox.

Images There are two scaling choices from which you can only select one, "Print actual size" and "Scale to fit page". If the image you are printing out is very large, you will have better success with the latter option. In most cases actual size will produce a better quality print.

26.2 Printing Index Cards

When viewing a collection of items on a corkboard, you can print the content of the corkboard onto index card sized blocks on a page. The formatting for this has been optimised to work with Avery(tm) Perforated Index Card stock, but you can use any paper with a chopping block or scissors to cut the cards apart if need be, or even feed in regular index cards if your printer supports abnormal paper feeds.

Since this method is optimised for printing to standard index card size, it will not print a perfect copy of what you see on the corkboard, and in most cases this would not be desirable anyway as the background textures and such would be a waste of ink, and longer synopses would only show what could be seen without scrolling. So instead, the content of the card's title and synopsis will be printed, and will continue to print on subsequent cards until the entire synopsis has been printed. Cards with lots of text content may take several cards to print out.

If the corkboard contains images, the image thumbnail will be placed into the card area unless that item has specifically been set to display the text synopsis instead of the image thumbnail. Note that most meta-data will not be shown when an image is placed into an index card as a thumbnail.

As with printing text documents, you can access options for corkboard printing in the File ► Page Setup... menu, navigate to Scrivener settings in the "Settings" drop-down, and then click the Options... button. You will find an "Index Cards" tab with the following

options. As you adjust these options, the preview pane above will change to reflect your choices.

- *Include titles*: adds the title of each card to the top.
- *Include card numbers*: this is similar to **View ► Corkboard ► Show Card Numbers**. Each card will be numbered relative to the corkboard scope, starting with 1.
- *Ignore cards with titles only*: with this option enabled, if a card has no synopsis, it will be ignored. This includes images that would otherwise be printed out as image thumbnails.
- *Highlight titles with label color*: the background of the title area will be highlighted with the card's assigned label colour, if relevant. The label name will also be placed in parentheses after the title, so if you are using a black and white printer, this information will still be available to you.
- *Include keywords*: when enabled, all keyword names will be printed out in a comma-delineated list below the title.
- *Print cutting guides*: this option is most useful when using standard paper. Each card will be outlined with dashed cutting guides, making it easier to separate them into actual cards with a cutting block or scissors. If you are using perforated card stock, it is best to leave this option off.
- *Force landscape orientation*: maximises the number of cards you can fit onto a single sheet of paper to four instead of three. If you are just printing to regular paper and plan on cutting them apart, use this option to save paper.
- *Embolden titles*: The title of the card, printed at the top, will be emboldened to set it apart from the rest of the card content.
- *Print using font*: override the default font with your preferred font family and size.

26.2.1 Tips for Printing to Individual Cards

With some printers, it is possible to feed individual cards into the printer, which makes for a cheap alternative to perforated cards. You will want to ensure that your printer is capable of handling thicker paper in small sizes before attempting this. Follow these steps to set up Scrivener:

1. Select the **File ► Page Setup...** menu item.
2. Under "Paper Size", select the menu choice, **Manage Custom Sizes...**

3. Press the + button to create a new custom paper size and call it “Index card”, or whatever you prefer.
4. Enter the height and width of your index cards. Use a ruler if you are unsure. Most cards come in 3 x 5 inch and 4 x 6 inch form.
5. For the non-printable area, you can select whatever you like here. A small value of 0.25 inches is a good default.
6. Click OK to confirm the new paper size and then make sure that is the selected paper size before clicking OK again to confirm your page setup.

You will now need to follow the instructions provided to you by your printer manufacturer to figure out how to feed the cards into the machine. Some printers will let you place the cards in a stack, but with many printers you will need to feed in a card one by one as it prints. Keep this in mind if you intend to print out hundreds of cards!

26.3 Printing Outlines

As with corkboards, you can print a limited range of content from an outliner view by simply viewing the material you wish to print in the outliner, and press ⌘-P. Also, as with corkboard printing, this will not attempt to reproduce the precise appearance of an outliner, as this would lead to situations that would be difficult to print, with very wide outliners. Instead, the outliner printing tool generates an indented list with the title and synopsis for each item in the current outline view, in addition to a few key meta-data fields.

Each row in the outliner will be printed in a “block” on the page, and within this block there are three primary sections in this order:

1. Title Area
2. Content Area
3. Meta-data Area

If you find the following settings for the outliner fails to provide the look you desire, you might consider using the compiler instead. Try starting with the provided preset, “Enumerated Outline” and customising that if necessary. Or, if you just want the data, try exporting the outliner to a spreadsheet compatible file (section 25.4).

Information will be placed into one of the three listed areas if enabled, and the following list of options will refer to these sections to indicate where the information goes.

To adjust the appearance of this print form, options have been provided in File ► Page Setup...; select "Scrivener" from the "Settings:" drop-down menu, and then click the Options... button, followed by the "Outlines" tab. As with outliner and text document setup, a preview will be provided as you change options, letting you know roughly what the final form will look like.

- *Include titles*: adds the title in the Title Area, using a bold font.
- *Include synopses*: the full synopsis will be printed for each item in the Content Area.
- *Include label and status*: if the label and status have been set for the row, they will be added to the section. Label will highlight the background of the Title Area with its associated colour, and will be inserted after the title (if present) in parentheses. Status will be placed at the bottom of the entry in the Meta-data Area.
- *Include keywords*: keywords will be added on the line above the synopsis, in the Content Area, in a comma-delineated list.
- *Include custom meta-data*: will be added in the Meta-data Area, one field per line for each meta-data field that has been filed out for that row. If a row does not have any custom-meta data assigned to it, then nothing will be printed here.
- *Indent by level*: indents each item relative to the current outliner view. So items which are children of the root level items will be indented once, no matter how deeply nested they are within the binder. Disabling this produces a flat list.
- *Print file name*: prints the name of the *container* that is being printed (which is otherwise not represented in the printout). If the outliner is the result of a multiple selection, then the label will simply be "printing". This will be placed in the header area of each page, on the right hand side.
- *Include word count*: the word count for that row (not the total word count, which would include its children) will be placed into the Meta-data area.
- *Include character count*: as above, only printing the character count instead.
- *Include targets with counts*: if any of the above two counters have been enabled, and a goal has been set for the row, it will be printed after the relevant counter.
- *Prefix titles with number*: if titles are enabled, the row number (relative to the outliner, not the binder) will be prefixed to the title. This is a simple incremental counter, not a hierarchal counter.

26.4 Printing the Draft

Printing the entire draft, the portion of your project that is set aside for your work in progress itself, is accomplished via the compiler, and is not unlike compiling to a saved file. You will use the same interface to print, and all of the options available to you during compilation will be functional. To print, open the compile interface with **File ▶ Compile** (⌥⌘–**E**) and using the "Compile For:" drop-down menu, select "Printing", then click the Compile button.

Once Scrivener has finished processing your draft, you will be presented with the standard OS X print dialogue. From here, you can choose to save a PDF from the drop-down menu on the left, or click the **Print** button to send the compiled document to your default printer. The first time you do this, you should Preview the print job, first. If you do not see a **Preview** button, make sure the print dialogue is in its collapsed state by clicking the little upward pointing arrow beside the printer selection menu. This will temporarily load the print job into Preview, and if everything looks okay, you can click the print button in the footer bar of the preview area.

See also:

- Read Compiling the Draft (chapter 24) for more details on how to compile.
- If you want to print a script, read Printing or Exporting a Script (section 20.4) for tips.

Part V

Advanced

I love being a writer. What I can't stand is the paperwork.

Peter de Vries

Chapter 27

MultiMarkdown Tips and Tricks

27.1 Functional Annotations and Comments

The functional usage of annotations is a largely advanced feature for MMD, and requires the usage of custom XSLT coding to take advantage of, and requires a little understanding of how notes are compiled into the MMD document.

For the most part, the HTML element that Scrivener uses is "span". Commented text will be wrapped in a span element, classed as "annotation" for CSS and XSLT matching purposes, and then styled using an inline CSS directive to colour the span in accordance with the RGB value used in Scrivener to codify the annotation or comment. Put another way, a red annotation will end up styled as red in your HTML document. MMD's default engine then takes that a step further with LaTeX, using the colour package to style text as coloured ranges in LaTeX. You can thus have commented text go all the way from Scrivener to LaTeX, appearing roughly as it does in Scrivener, without the bubble.

An alternate form that Scrivener will use is when inline annotations are on paragraphs all by themselves. In this scenario, a "div" element will be used to wrap the entire section. For those versed in HTML, the difference between the two will be readily apparent, but for the rest, this basically means that the content of a "island comment" will be capable of *structural detail* in a way that an "embedded comment" cannot be. An embedded comment couldn't utilise paragraphs within the comment, because that would disturb the document flow from its original appearance, and it would also violate the HTML specifications as paragraphs cannot exist within span elements.

In practical terms, this means you can use full MMD syntax within so-called "island comments", something you cannot do in embedded comments, you can only use simple formatting like italics and bold—anything that does not require a block element (like a paragraph, or a bullet list) to express. In a sense, this elevates island comments to a fully functional role. Using a custom XSLT which is designed to look for them, they can be used to ascribe additional semantic meaning to a document at the block level. This can also be done for embedded comments, but only so long as the intended syntax does not require a block level expression.

The tip boxes in this PDF are an example of these so-called "island comments". Prior

to this transition, it would have required me to type HTML div coding directly into the Scrivener project in order to create special paragraphs like the tip box—or I would have had to have designed some syntax and gone through all of the steps required to add that syntax to MMD (something which should be avoided if at all possible). Instead, I can simply mark out a section of my document in an annotation, and prefix that annotation with the phrase "TIPBOX: ", which the XSLT has been designed to look for, discard, and then format the div block in such a way that it is displayed in a LaTeX "leftbar" environment.

Due to their nature, inspector comments have no div element form, as they will always at some level be embedded within another paragraph.

27.2 Multiple Footnote Streams with MMD

Each type of footnote will be exported using a different reference prefix. This prefix is just a text-based convention, and is in the default MMD installation, completely devoid of meaning. Inline footnotes will be exported with a "fn" prefix, followed by a sequential number which is incremented for each footnote that is processed. Meanwhile, linked inspector footnotes will be prefixed with "cf", instead. In the standard export workflows, this will actually be completely invisible to you. Both will come out looking identical, if you mixed types within Scrivener. However because a prefix has been used to denote each type, this gives you the opportunity to apply additional rules to your custom XSLT which handles each note type as a separate stream. If your work requires both footnotes and endnotes, it would thus be possible to generate an XSLT that produces these, provided the final output format it is generating can handle it.

Chapter 28

Customising Keyboard Shortcuts

You can customise shortcuts in OS X's Keyboard preference pane. Just click on the "Keyboard Shortcuts" tab in that pane, and then "Application Shortcuts" in the sidebar. For example (to add a shortcut to the Show Collections) follow these steps (Snow Leopard and greater):

1. Click the + button below the right list
2. Select Scrivener in the "Application" drop-down
3. Type in "Show Collections" for the menu title. This part is important, the name of the menu must be typed in completely and precisely, including capital letters, and the three dots at the end if applicable, though you do not need to type in the existing shortcut if it had one.
4. Click into the keyboard shortcut field, and then press the keys you wish to use for this shortcut. If you accidentally press the wrong keys, you can try again by clicking back in this field and pressing the right combination.
5. Click the "Add" button.

Now switch back to Scrivener and test the shortcut. In some cases you may need to restart to program for menu shortcut changes to take hold.

Part VI

Appendices

Appendix A

Menus and Keyboard Shortcuts

This appendix will address each item in every menu, and will display the keyboard shortcut for it when applicable. It is intended to be a quick, exhaustive reference of the menus, and will attempt to point you in the right direction for more thorough discussion and analysis of various features when necessary.

The menus in Scrivener are divided into various domains, in which we attempt to organise functions into them according to those domains. If you are looking for a particular feature and are not sure where it is located, then reference the table describing the menu sections (Table A.1). In this way you can try to track down a feature as you might in a topical thesaurus. If you happen to know the name of the feature you are looking for, and have access to the software, the easiest way to find a menu command is to use the search bar from the Help menu.

See also: Scrivener Shortcut Reference[1], on the Web.

A.1 Scrivener Menu

Much like the application menu in any other Mac OS X program, this provides access to application level information, features, and system integration, such as Services.

About Scrivener Displays the credits and version number of the application. If you are experiencing problems and wish to contact customer support regarding them, you can provide version information using this dialogue. Simply click anywhere outside of the scrolling text area to dismiss it.

Preferences... (⌘-,) Accesses the main application preferences window. For a complete list of all available options, see Preferences (Appendix B).

Reveal Support Folder in Finder [10.7+ (Lion) Only] Opens Scrivener's "Application Support" folder, where it stores your automatic project backups, custom project templates, custom icons, and so on. Use this if you wish to transfer settings between machines, or if asked to do so for troubleshooting reasons.

[1] http://scrivener.tenderapp.com/help/kb/mac-os-x/keyboard-shortcuts

Table A.1: Menu Organisation

Menu Title	Purpose of Menu
Scrivener	The main application menu is where overall application settings and information can be accessed. Changing preferences, setting up registration information, and quitting the software can be done from here.
File	Concerning the management of projects, including the transfer of information into and out of them, either manually with import & export, or automatically with backups and synchronisation.
Edit	If a function will work with the *content* of an editor window, either passively or to manipulate it directly, then chances are it will be located in the Edit menu. Examples would be copying text, inserting an equation, checking spelling, or searching for all bold phrases in the text.
View	For adjusting how the project window looks and feels, as well as navigating (or changing what you are viewing) within the content of the project. The features here are not strictly aesthetic, and many work directly with project content itself, such as the View ► Go To ► sub-menu. Other features are simply concerned with the control of window features, such as View ► Media ► Rewind, or the appearance of things, as in View ► Corkboard Options ► Cards Across ► .
Project	Settings and features specific to the project are gathered in this menu. This is also where content can be created, and is distinct from the File menu in that the File menu is concerned with creating new content in the project based off of existing data on your disk (such as importing a picture). The Project menu is for creating brand new content.
Documents	This menu concerns the management of existing documents, as well as direct navigation based off of a selection of documents. It is distinct from View menu navigation, which is concerned with more abstract navigation, "Go to this location here…", instead of, "Open this document I have selected…". Various aspects of a document's meta-data and organisation can be set in this menu.
Format	Where the Edit menu concerns itself with the *content* of an editor session, the Format menu concerns itself with the *presentation* of content. In simple word processing terms, this includes such minutia as setting a range of text to bold or italic, but it also includes settings of larger impact, such as whether paragraphs should be numbered, if titles should be included in Scrivenings mode, and the ability to add footnotes or comments to the text.
Window	Closely related to the Scrivener menu, in that it operates on a level above projects. A list of all open windows will be provided, as well as tools for re-opening recently closed QuickReference Panels, and the familiar window controls for minimising or zooming the size of a window.

Registration... [**Standard Version**] When you purchase the application, use this menu item to copy and paste your registration information from the e-mail that you will receive from Literature & Latte. In case you have lost the original e-mail or never received the invoice, you can use the **Retrieve Lost Serial...** button which will take you to a web page with further instructions on how to retrieve it. Not infrequently, invoices can end up in spam collection folders, so be sure to check your e-mail application and make sure it did not get tagged as spam, first.

Check for Updates... [**Standard Version**] You can use the Check for Updates menu item to check to see if there is a newer version of Scrivener available for download (2.x updates are free). This tool lets you check for updates and then download and install them all without leaving the application.

Services ► The items in this menu are provided by the core system, and other third-party applications. They will let you perform various functions, mostly based on selected text. Scrivener provides its own services which are also available in this menu, and from other applications as well. See Scrivener Services (section 11.2) for further documentation.

Quit Scrivener (⌘-Q) Leaves the program. Any projects that are currently open will be saved to the disk, if necessary, and backed up for you. Under the default preferences, these projects will be remembered and opened automatically the next time you run Scrivener.

Quit and Close (Keep) Windows (⌥⌘-Q) [**10.7+ (Lion) Only**] Whether this displays "Close" or "Keep" windows depends upon whether your preferences are set to re-open projects that were open on quit (subsection B.2.1). Which is displayed will be the opposite of whatever your standard settings are. This will not change your preference; it only impacts the one time you quit.

A.2 File Menu

The File menu contains everything that handles creating files on your computer, including creating new projects, saving, backing up, importing and exporting. It also deals with printing and project-specific settings.

New Project... (⇧⌘-N) New Project creates a new Scrivener project. This will bring up the Project Templates window (see Quick Tour (chapter 6)) which will walk you through creating a new project file in the location you specify.

Open... (⌘-O) Open allows you to open a pre-existing .scriv file on your hard disk. Note that Scrivener cannot open read-only files directly (such as those that have been backed up to a CD-ROM) and will give you the opportunity to save the project to an area where you have permission to save files.

Recent Projects Here you can select and open projects that have recently been opened in Scrivener (you can also specify in the General Preferences (section B.2) whether Scrivener will reopen all projects that were open in a previous session when it is launched).

The number of items listed in this sub-menu is governed by OS X's global preference, which can be set in the system Appearance preference tab.

Find All Projects in Spotlight Runs a Spotlight search, bringing up a Finder window with the results for all Scrivener projects included within the remit of the Spotlight index. It is important to distinguish the difference between this and searching your entire computer for projects. The Spotlight index may not include drives you have specifically omitted, or external drives you have plugged into the computer.

Close Project (⇧⌘-W) Closes the current project and all of its associated windows.

Close Project and Clear Interface Settings (⇧⌥⌘-W) When holding down the Option key and viewing the File menu, you'll see this mechanism for closing a project. This will safely shut down the project and as a final step, trash the project's "UI" preference file. This means the project will go back to how a brand new Blank project looks. All options such as interface component visibility (rulers, inspector, etc), splits, label tinting, columns, and other settings will be factory reset. Ordinarily, you'll only ever need to do this as a troubleshooting step, or a way of fixing a layout glitch you've encountered.

Close Window (⌘-W) Closes the currently active window, QuickReference, or dialogue box. However if the project window has focus, the entire project will be closed instead. Composition mode must be exited explicitly, it does not respond to ⌘-W.

Save (⌘-S) Scrivener auto-saves your writings as you work, so that you never have to worry about losing your efforts. Projects are also saved automatically whenever they are closed. However, you can use Save to force an immediate save whenever you want.

Save and Rebuild Search Indexes (⌘-⌥S) When holding down the Opt key (⌥⌘-S) you can manually rebuild the search index. This is sometimes useful in cases where you suspect the project has lost synchronisation with search index. There is generally no need to do this as Scrivener will automatically rebuild indexes if something appears to have gone awry since the last session.

Save As... (⇧⌘-S) Will prompt you for a new project name and/or location. When you submit this dialogue box, Scrivener will immediately start working off of the **new** copy. If instead you want to generate a backup copy and keep working off of the original, use File ▶ Back Up ▶ Back Up To... instead.

For more information on managing your project files, see Project Management (chapter 7).

Import ▶ Features for importing a variety of existing information, from your drive or even from the Internet, into your active project. For full documentation on importing information into Scrivener, refer to Gathering Material (section 11.1).

Files... (⇧⌘-I) Import files from the file system into Scrivener. Only text formats can be imported in the Draft folder. See the prior referenced section for a full list of supported formats there, and elsewhere in the Binder.

Web Page... (⌥⌘-W) Import and archive a web page from the Internet by supplying Scrivener with its URL.

Research Files as Aliases... Rather than fully importing items into the project, this command establishes a link between the original item and the project. This link will automatically adjust if the original item is moved or renamed. Read Linking to Research Material (subsection 11.1.3) for further details.

MultiMarkdown File... Takes a plain-text MultiMarkdown formatted document and will convert its header structure into a binder outline structure.

Plain Text Formatted Screenplay... Import a plain text screenplay, optionally splitting sections into separate binder documents. This works best with scripts that have been exported from CeltX and Movie Magic Screenwriter.

Scrivener Project... Import a selected Scrivener project into the current project's binder. All binder items will be imported, as well as some meta-data: keywords, references, and notes.

Import and Split... Takes a standard plain or rich text file and allows you to supply a character sequence (such as "#") which will be used to split the document into sections.

Export ► Provides tools for exporting elements of the binder to the file system. For more information on exporting your work, see Exporting (chapter 25).

Files... (⇧⌘–E) Allows selected contents of a project to be exported as files and folders onto the hard drive. This feature should primarily be used to create a hard copy backup, or to share resources with someone not using Scrivener. Generally speaking, you will want to use Compile to export your book.

OPML File... Exports the current binder selection as an Outline Processor Markup Language file. OPML files are commonly read by applications that work in outline oriented data, like Scrivener, and is thus a convenient way of transferring hierarchal information between programs.

See also: Exporting to an Outliner with OPML (section 25.3).

Outliner Contents as CSV When an Outliner view is active, this menu command will activate. It will export the current Outliner view in a format that is readily accessible to spreadsheet software and other programs capable of tabular display.

See also: Exporting Meta-Data to a Spreadsheet (section 25.4).

Comments and Annotations... Exports just the comments and annotations from the project into a single file. Optionally you can choose to export only the pre-selected binder items.

Sync ► Tools for linking parts of your project to external applications, mobile devices, or disk-based files and folders. Read more about syncing in Cloud Integration and Sharing (chapter 13).

with External Folder... Method for creating folders with plain-text, rich text, or Final Draft files on your system. This export location can be anywhere, including network mounted drives or synchronised folders such as provided by iCloud, Dropbox and SpiderOak.

with External Folder Now Simply runs the external folder sync tool immediately without loading the configuration window. If External Folder sync has not been previously set up, it will load the configuration window for you.

See also: Synchronised Folders (section 13.2).

with Simplenote... Method for synchronising individual files or folder content as plain-text to the Simplenote server, which can be used with a variety of applications and devices. Read more in Simplenote (section 13.1).

with Index Card for iOS... Index Card[2] for iPad is an application which lets you view items on a corkboard in a manner similar to Scrivener's corkboard. Using this tool, you can sync a collection (section 8.4) from Scrivener to a special .plist file that Index Card can use to import your material. Read more in Index Card for iOS (section 13.4).

Back Up ► Functions for managing backup copies of the current project. For more information on back up strategies, see Backing Up Your Work (section 7.11).

Back Up To... Generates a backup copy of the project to a specified location. Backup copies, unlike "Save As", will be created and then ignored by Scrivener, you will keep working in the current version of the project. Optionally, you can choose to compress the backup in a zip archive, which takes longer, but is the recommended method for storage of backups on network drives.

Back Up Now Triggers the automatic backup system to produce a backup immediately, using the preferences for backup location and rotation scheme. Customise how this works in the Backup preferences tab (section B.11).

Exclude From Automatic Backups Removes the current project from the automated backup system. This is useful when the current project is very large, and is causing you to have to wait for long intervals while the automatic system performs backups. Note that when this option is engaged, Scrivener's backup protection will be completely ignored for this project. Take care to manually back up excluded projects from time to time.

Save As Template... Saves the current project as a template. This will add it to the New Project window for use as a basis for your future projects. For more information on creating templates, read Templates (section 7.10).

For full documentation of the printing process, read Printing (chapter 26).

Page Setup... (⇧⌘–P) Accesses the standard page layout setup pane. Additionally accesses Scrivener specific features from the **Settings** drop-down menu. **Note**, these settings do not adjust Compile printing, only current document printing, though Compile can be optionally configured to use your Page Setup paper dimensions.

---

[2] http://www.denvog.com/iphone/IndexCard/

Print Current Document... (⌘-P) Prints the current editor view. How this will be printed depends on the view mode, and all of the settings for this can be accessed with File ▶ Page Setup... above.

Compile... (⌥⌘-E) Compile is the standard method for producing a manuscript out of all the pieces in the Draft. This feature provides an immense degree of flexibility, and is fully documented in Compiling the Draft (chapter 24).

A.3 Edit Menu

The Edit menu contains options related to editing. All of the standard Mac Edit menu items (with which you will be familiar from such programs as TextEdit) can be found here, including cut, copy, paste and find, alongside with a large complement of Scrivener-specific features.

Undo & Redo

Undo & Redo (⌘-Z & ⇧⌘-Z) Undoes or redoes the last change. Undo and Redo work mainly for edits made to text, but they do work for some basic outlining changes too, where that change does not cause the current document to change in the editor. Each document has its own Undo history, which means you can easily go back to another document and revert changes independently of other documents.

Cut, Copy, and Paste

Cut, Copy, and Paste (⌘-X, ⌘-C, and ⌘-V) Cut, copy and paste act exactly as they do in other applications. Note that they only operate in a text editing context, and not on binder documents or meta-data. See Documents Menu (section A.6) for document management commands.

Copy Special

Copy Special ▶ This menu allows you to copy text or binder items in a variety of ways which can apply or remove data depending on the specific menu command.

Copy as HTML Reproduces most formatting as HTML codes using inline CSS. The resulting HTML should look very similar to the text you copied in Scrivener, including some ruler settings, colour, highlight, and so on.

Copy as HTML (Basic using <p and <span) Applies minimal inline CSS formatting. Ruler styles and some types of formatting will be lost. This is often

the best choice for pasting into blogging and Content Management Systems, which provide their own stylesheets.

Copy as HTML (Basic using <br >) Applies very basic, HTML 4.01 compliant code. When constructing HTML e-mails, or working with an older web site, this is what you will want to use.

Copy as BBCode Translates most basic formatting into BBCode, suitable for forum posts. Because BBCode does not support as many formatting features as full HTML+CSS, some elements will be lost.

Copying without Comments or Footnotes (⇧⌥⌘-C) will strip out all annotation and footnote text, while retaining all other formatting. Most often this is useful for producing "clean" snapshots (section 15.6) after an editing session. This command will remove all notation, including linked notation.

The following menu commands operate on *selected documents*, not text:

Copy Documents as Scrivener Links Ordinarily, when you copy a selection of documents, pasting them will just list the names of the documents. This command will also build links back to each of the respective documents.

Copy Documents as External Link Produces a URL which can be used to link to the selected items from any other software supporting links, on your computer. You could for example paste this URL into the hyperlink URL field of your word processor, creating a link from that document to a particular item in your project binder. These links are static references to the project's location on your disk. They will not work from other computers, or if the project has been moved from the location it was in when the link was generated.

Copy Documents as ToC Will create a basic Table of Contents out of the selected documents, which is meant to be pasted into a Scrivener document. See Creating a Table of Contents (chapter 23) for further information.

Paste and Match Style

Paste and Match Style (⇧⌥⌘-V) This pastes the contents of the clipboard without any of its existing fonts and styles, in essence, treating it like plain-text. Useful for when you have copied a range of formatted text but want to paste it using the style of the text into which you are placing it.

Delete (⌘-Delete) Moves the selected binder items to the Trash. Scrivener works in the same fashion that the Finder does. Items are not directly deleted, but first moved to a Trash folder, located in the binder. Periodically, you can review the contents of this and empty the trash to clear up space.

Select All (⌘-A) Selects all of the content of the current text, outliner view, table view or corkboard.

Select Subgroups Applicable only to the Outliner and the Binder when a container of any type is selected, this command will reselect any descendant container items that are currently visible. For example, if a folder has three other folders beneath it which are collapsed and one text document, this command would select the three collapsed folders. If one of those folders was disclosed and it contained file group within it, then the three folders and that 3rd level file group would be selected. This command can come in handy when assembling corkboard stacks (subsection 12.1.6).

Select Current Text (⌥⌘-A) When holding down the Option key in the text editor, this alternate method will be displayed. It is useful for selecting only the text in the section you are currently editing within a Scrivenings session, rather than the entire session.

Select Group with Descendants (⌥⌘-A) When used in the Outliner or Binder, this alternate method will be displayed. It provides a shortcut to select an item and all of its descendant items at once. This can be useful when performing an action that requires a literal selection. Corkboards cannot use this function, as they can only display one level of hierarchy at a time.

A.4 View Menu

The View menu contains all of the commands related to changing the way documents are viewed, allowing you to show and hide various elements, navigate between views and customise the way information in the current project is displayed. A rule of thumb is, if you want to change the way something in your project looks or acts, and it's not a setting in the main options, chances are it is in the View menu.

View Modes are various ways in which you can view the presentation of multiple items in the editor. For and overview of how view modes work, see View Modes (section 5.2).

Document/Scrivenings (⌘-1) If multiple items are selected in the editor, displays the current editor as a Scrivenings text editing session, showing the text of each document as if they were a single file. Also sets your preferred collection view mode to Scrivenings. Note this option is only useful if the editor is displaying documents, media will not be mixed into an Scrivenings session.

If a single item is selected in the editor, or if the editor is currently displaying a Scrivenings session, this menu item will be labeled **Document**, and switch the editor focus to its text or media view.

See also, Editing Multiple Documents (section 15.9).

Corkboard (⌘-2) Display the current editor as a corkboard. If the selection is a container, its children will be shown as index cards. An fresh, empty corkboard will be displayed otherwise. Also sets your preferred collection view mode to Corkboard. See also, The Corkboard (section 12.1).

Outline (⌘-3) Display the current editor as an outline. Also sets your preferred collection view mode to Outliner. See also, The Outliner (section 12.2).

The menu items, "Go To" and "Text Bookmarks" sub-menus are also available from the header bar icon menu (subsection 14.1.2).

Go To ► Offers an alternate way to jump to a specific binder item without using the binder. If the binder is hidden, a collection is selected, or a section is hoisted, for instance, will let you navigate to other areas of the project without having to alter your work environment. Also provided are some contextual navigation and focus functions which will let you move around in the binder, from the editor itself.

Holding down the Option key while selecting an item from this menu will cause it to be opened in a QuickReference panel, instead of changing the current editor.

- *Previous Document*: (⌥⌘-UpArrow) Jumps to the previous document in the binder, according to strict outline order (ignore hierarchy).
- *Next Document*: (⌥⌘-DownArrow) As with Previous Document, but selects the next document in the strict outline order.
- *Enclosing Group*: (^⌘-R) Will display the currently edited item in context with its siblings using the current view mode. In most cases, this will select the immediate parent of the document, selecting the document's index card or outliner row that you came from, for your convenience, or expand your current Scrivenings session to include the siblings and parent of the current text item.
- *Editor Selection*: (⌘-4) jumps directly to the text component of any selected item. This works as an isolation feature in Scrivenings, or can also be used to quickly open selected item(s) from Corkboard or Outliner as text (even if they are containers). The difference between this command and ⌥⌘-O or ⇧⌘-O

is that these latter two will respect the typical view load priority setting. In other words, if the selection is a container, these commands will load that item in a view. ⌘-4 will *always* load the selected item as a text file. When more than one item is selected, they will be loaded as a composite Scrivening in Multiple Selection mode.

QuickReference ► Similar to the Go To menu, except that the selected item will be opened in a QuickReference panel, rather than replacing the contents of the editor. The contents of this menu will be populated by the binder in all cases.

Collections ► Provides commands for setting visibility and options of the Collections feature. For documentation on how to use Collections, see Using Collections (section 8.4).

Show (Hide) Collections Reveals or hides the collection interface which will appear above the binder. Note the sidebar will need to be visible for this menu option to be available.

Convert to Standard Collection When a Search Result Collection is the active tab, this command will "freeze" the results and turn it into a regular collection that is no longer dynamically updated. This is a one-way process that cannot be undone. To freeze a search result list without converting the original collection, simply select the entire list of items and add a new collection. The selected items from the search result will be added automatically to this new collection for you.

Next Collection Select the next Collection tab in the stack

Previous Collection Select the previous Collection tab in the stack

---

The rest of this menu will be populated by the existing available collections in the reverse order that they appear in the tabs. It will always include an entry for the binder and Search Results. Selecting an entry will switch the sidebar to viewing that collection, making this menu useful when the tabs are not visible, or for assigning keyboard shortcuts to oft-used collections.

Text Bookmarks ► Jump to places in the current document which have been bookmarked. To insert bookmarks, see Edit ► Insert ►. The contents of this menu will be populated by existing bookmarks from the current document or combined in a longer list, when using a Scrivenings session.

Zoom ► Provides shortcuts to the zoom menu, located in the footer bar for text documents. Temporarily zooms the display of the font up or down to increase legibility, without adjusting the underlying font size. This will also increase or decrease the size of a viewed PDF. When the Inspector footnotes and comments pane is active, this menu will scale the size of all notation text, instead.

- *Zoom In*: (⇧⌘–<) makes the text larger.
- *Zoom Out*: (⇧⌘–>) makes the text smaller.

Layout ► The commands in the Layout sub-menu affect the main window elements.

Show (Hide) Binder (⌥⌘–B) Toggles whether or not the binder (subsection 5.1.2) sidebar is visible on the left of the main window. You can hide the binder to concentrate on editing or composing the current document if you so wish.

Show (Hide) Inspector (⌥⌘–I) Toggles whether or not the Inspector (subsection 5.1.5) is visible on the right of the main window. The inspector displays all meta-data for the current document, including synopsis and notes. The inspector is hidden by default.

No Split (⌘–'); Split Horizontally (⌥⌘–=); Split Vertically (⇧⌘–") These commands allow you to choose how to view the main editor in the main window. For more information on using splits, see Splitting the Editor (section 14.4).

Swap Editors Only available when Split Horizontally or Split Vertically is selected (see above). This commands swaps the position of the two split panes. For example, so that the view on the top will become the view on the bottom and vice versa.

Show (Hide) Header & Footer View Toggles the visibility of the header and footer bar. Both are elements of the editor and can be toggled independently for each split. See The Editor (chapter 14) for more information on what these do.

Show (Hide) Layouts (⇧⌘-) Toggles the Layouts panel, which allows you to manage view settings by saving the current settings, or retrieving other settings you've saved in the past. See Layouts Panel (section 9.7).

Editor ► Options and commands for controlling the main editor.

Lock in Place (⌥⌘-L) Locks the editor (or split) so that no binder clicks will affect it. When an editor is locked, its header bar will turn a shade of red.

Lock Group View Mode Active only when the active editor is displaying a group view, or if the selected item is a container (folder, or file group acting as a folder). Any view mode can be locked, including single text document mode. Locking is specific to the active container. The view mode currently used while it is locked will remain what this container uses to display itself even if you change your preference later to another mode. See Locking the View Mode (subsection 9.4.1)

Scroll to Next/Previous Page These menu commands will only activate when Page View is enabled on the active editor. They will flip the scroll bar by page, rather than by relative position, as the keyboard PageUp and PageDown keys do.

Forward/Backward in Document History (⌘-(and ⌘-)) Like a Web browser, the editor keeps track of everything you've visited. Using these commands you can navigate back and forth in the history. Note you can also use the Forward and Backward buttons in the header bar. Each split keeps its own history.

When selecting a number of documents in the binder prior to entering composition mode, the history queue will be pre-stocked with these selected items, so long as the main editor is not set up in Scrivenings mode.

Other Editor ► Provides controls to manipulate whichever split is not currently active.

- The Forward and Backward in History commands (⌥⌘-) & ⌥⌘-)) operate just like the above commands, only in the opposing split without moving focus there.
- The Scroll Up and Down commands (⌥⌘-UpArrow & ⌥⌘-DownArrow) act like PageUp and PageDown, only in the opposing split, again without moving focus there.

Clear Document History Use this command to wipe out the document history for the active editor.

Page View ► Accesses features and options for displaying the text editor as a virtual page, instead of filling the entire editor. It is not intended to be used as a layout mechanism, though it can provide a reasonable estimate of how your pages will look under certain conditions, once printed. Read more in Page View (chapter 16).

Show (Hide) Page View (⇧⌥⌘–P) Toggles Page View on and off.

Two Pages Across When enabled, instead of displaying pages in vertical column, two will be placed beside each other as though reading a book.

Media ► Controls and options for various types of media. These commands will only become available when viewing the appropriate type of media.

*** PDF Display ►** This sub-menu controls how PDFs are displayed in the editor. Most of these options should be familiar as they are common to many PDF viewers.

- *Automatically Resize*: Keeps the PDF sized to the width and height of the editor window.
- *Actual Size*: Zooms the PDF to 100%, even if it is too large to display all at once.
- *Size To Fit*: Like "Automatically Resize", but only resizes the document once. If you change the size of the editor view later, it will stay at the same view.
- *Single/Facing Pages*: In single mode, will show one page per row like an ordinary digital file. In facing mode, two pages will be placed side-by-side in a column, more like reading a book.
- *Continuous/Page Breaks*: These two scrolling modes affect how the document is handled when using the scroll wheel, arrow keys, or page up and down keys. In Continuous mode, the pages flow by seamlessly; in page break mode, only one (or two, if facing pages is enabled) pages will be showed at a time, and scrolling actions flip between pages.

*** Play Media File (⌘–Return)** Functionally, this acts like clicking the Play button in the media viewer, for either audio or video files. If the editor is split between a text document and a media file, and you are typing in the text document, this shortcut will start and stop the media in the *other* split, allowing

you to easily transcribe or reference the media. When two media files are open at once, the shortcut will affect the active split.

* **Fast Forward/Rewind (⇧⌥⌘-) / ⇧⌥⌘-(** Will jump playback forward or backward by two seconds. This shortcut works while playback is running, or while paused. In the latter case, it will stay paused at the new position. As with *Play Media File*, will work in the opposing split while typing, providing further aid for transcription.

Corkboard Options ► Features and visual options for the corkboard. The settings in this menu are project specific, and some of them will only impact a single split, allowing you to have different view settings for each editor. For more information on using the corkboard, see The Corkboard (section 12.1). Note that further options can be accessed via the corkboard display options button, which is located on the right-hand side of the footer bar for each corkboard.

See also, Setting Up Meta-Data (chapter 10), for more information on the types of meta-data that can be viewed on an index card.

Cards Across ► The Cards Across submenu allows you to define how many index cards you would like to appear in each row on the corkboard. The default is three. Auto-fit will calculate how many cards to show, based on the size of the editor and the size of the cards; kind of like word wrap, for index cards.

Show (Hide) Pins (⌃⌘-P) Toggles whether the coloured pins are displayed on the index cards. Pins take on the colours associated with the labels that have been assigned to the documents represented by the index cards. If you are using the Rounded Card theme, pins will be displayed as colour swatches on the upper right side of the card. The appearance of pins can be adjusted in the corkboard tab (section B.4) of preferences.

Show (Hide) Stamps (⌃⌘-S) Toggles whether stamps are displayed on index cards. Stamps show the current status associated with the document represented by the index card as though it has been "stamped on" the index card. The appearance of stamps can be adjusted in the Corkboard tab (section B.4), and the Appearance tab (section B.3).

Show (Hide) Keyword Colors (⌃⌘-K) Keywords assigned to a document can be visually indicated along the right-hand side of the index card as coloured pieces of tape. The number of keywords that can be shown at once can be changed using the corkboard display options button. For more information on using keywords, see Setting Up Meta-Data (chapter 10).

Show Card Numbers Cards will be numbered according to their sequence in the binder outline structure. Ordinarily, this will match the order in which the cards appear, but in Freeform mode, this can be a valuable tool as the order of cards are likely to become desynchronised with the actual manuscript structure.

Number Per Section When more than one container has been selected, and the corkboard is displaying cards in Multiple Selection mode, card number will be re-started for each new section. Otherwise, cards will be numbered in the order they appear from start to finish.

Freeform Activates the Freeform Corkboard mode, which allows you to move cards around on the corkboard without any grid or ordering constraints. This can also be toggled in the Footer bar.

See also: Freeform Corkboard (subsection 12.1.4).

Commit Freeform Order It is possible to instruct Scrivener to use the current freeform layout to re-order the actual manuscript outline structure. Scrivener will provide several options for how this should be done. For more information on how to work with Freeform Corkboards, see The Corkboard (section 12.1).

Outliner Columns ► Displays a list of toggles which will reveal or hide the corresponding columns in the active Outliner view. These settings are stored by individual editor split, letting you have custom columns for the two views, and are saved in the project file. You can also manage columns using the "►►" button above the scrollbar in the outliner itself.

See also: Managing Columns (subsection 12.2.1).

Use Label Colour In ► A document's label colour (see Setting Up Meta-Data (chapter 10) for more information on labels) can be applied as a tint to various interface elements throughout Scrivener. This menu contains a list of toggles which will enable tinting to the corresponding elements. The intensity of all background tinting can be adjusted in the Appearance preferences tab (section B.3).

- *Binder*: Rows in the binder will have a background bar placed behind items with labels in a fashion similar to how Finder applies labels to files.

- *Icons*: Only the icons will be tinted throughout the project. Wherever the icon for a binder item shows up (such as in the Header bar, next to the title in Corkboard, in the binder, and so forth) it will be tinted using the colour of the assigned label.
- *Index Cards*: The entire background "paper" for index cards will be tinted using the assigned label colour. This includes the index card which appears at the top of the inspector.
- *Outliner Rows*: As with binder tinting, the background for the entire row will be filled in with the label colour.

Move Focus To ► Provides application focus navigation tools. Rather than navigating around in your project, these commands will let you quickly select different parts of the application without using the mouse. The first lets you cycle between common elements, while the rest will jump immediately to that element of the interface, no matter where focus is currently placed. Note that in all cases, the elements you wish to cycle or jump to must actually already be visible. These shortcuts will not automatically reveal parts of the interface that are currently hidden.

- *Left/Right/Top/Bottom Editor & Binder*: (^Tab) This menu item follows a chain of the three most common areas of desired focus, the primary split, secondary split, and the binder. It will cycle between these three going left to right, and the label of this menu item will be changed to indicate where the next target will be.
- *Binder*: (^⌥⌘-B) Moves focus to the binder sidebar from anywhere in the interface, if visible.
- *Left/Bottom Editor*: (^⌥⌘-E) Moves focus to the Left or Bottom editor, if the editor has been split. Otherwise selects the primary editor if there is no split.
- *Right/Top Editor*: (^⌥⌘-R) Moves focus to the Right or Top editor, if the editor has been split. If it has not been split, this shortcut will not be available.

Inspect ► A continuation of the navigation functions, these deal solely with the revealing and focussing of elements within the Inspector. All of them work on the following principle: if the keyboard shortcut *revealed* the item (it was hidden before), then the action will simply reveal the meta-data without disturbing your original focus. If the item is already revealed, then the shortcut will *switch focus* to that meta-data area. If the Inspector is hidden, it will be opened if necessary, which counts as a "reveal" action.

- *Synopsis*: (^⌥⌘-I) Reveals or switches to the index card in the inspector (will use the last inspector pane coupled with the synopsis and meta-data panes, if necessary).
- *Notes*: (^⌥⌘-H) Reveals or switches to the notes editor within the inspector.
- *References*: (^⌥⌘-N) Reveals or switches focus to the references table. * *Keywords*: (^⌥⌘-J) Reveals or switches focus to the keywords table. Arrow keys can be used to navigate amongst keywords; Return can be used to add new keywords; and Delete to remove keywords.
- *Snapshots*: (^⌥⌘-M) Reveals or switches focus to the snapshot text area, allowing quick copy and paste of its contents.
- *Comments and Footnotes*: (^⌥⌘-K) Reveals or switches focus to the comment list. Arrow keys can be used to navigate amongst contents; and expand and collapse notes.

Outline ► Provides navigation and disclosure commands which work in both the outliner and in the binder, referred to below as the "active outliner". Container is defined as a folder by default, but if "Treat all documents with subdocuments as folders" is enabled in Navigation preferences (section B.6), the definition of container will be expanded to include document stacks as well.

- *Expand All*: (⌘-9) Expands all collapsed items recursively. When this command is used in the binder, it will expand the entire binder (including all non-Draft sections). When used in outliner, it only works within the context of the currently viewed portion of the binder. This command can also be used in the Footnotes & Comments Inspector sidebar to expand all notes to full height.
- *Collapse All*: (⌘-0) As with Expand All, this works the other way, closing all visible open items recursively. Note that in both cases, you can use the Option key in conjunction with a mouse click to *selectively* perform this function on the arrow (and everything beneath it) clicked. This command can also be used in the Footnotes & Comments Inspector sidebar to collapse all notes to single lines.
- *Collapse All to Current Level*: (^⌘-0) Working in a similar fashion to Collapse All, this command will collapse the entire outline only up to the currently selected item's level.
- *Previous Container*: (^⌥UpArrow) Will select the previous visible container in active outliner, no matter what the depth. In this case, previous means a container higher on the list than the current position. If a container is currently

hidden, it will be skipped. If you only wish to select the parent container of a the current item, you can use the LeftArrow key.

- *Next Container*: (^⌥DownArrow) Works the same as *Previous Container*, only selecting the next visible container downward from the current position.

Reveal in Binder (⌥⌘-R) This command is available the editors and any collection view in the sidebar. It will show the location of the currently edited (or selected) file(s) in the binder, opening containers as necessary to do so. This is most useful when the method you used to arrive at the current document did not involve clicking in the binder. This menu command is also available from the header bar icon menu.

Binder Affects ► Adjusts which split binder sidebar clicks will be opened in. By default all clicks will be sent to the active editor (so long as it is not locked). This will adjust that behaviour so that the clicks go to a predetermined editor in all cases, depending on the choice made below. If the editor is not split, this menu will be deactivated as all clicks will naturally go to the only available editor.

When any of the following options besides "Current Editor" is selected, a small icon will be placed at the top of the binder sidebar. This icon will provide a visual aid in demonstrating which editor is currently receiving clicks.

- *Current Editor*: Default behaviour as described above.
- *Other Editor*: Whichever editing split is *not* current active.
- *Left/Top Editor*: The left/top editor no matter which editor is active.
- *Right/Bottom Editor*: The right/bottom editor split, no matter which editor is active.

Enter (Exit) Full Screen (^⌘-F) [10.7+ (Lion) Only] Utilises Lion's full screen implementation, expanding the main project window to occupy the entire screen. In this mode, the project window will be moved to its own "Space[3]", and cannot be resized until Exit Full Screen has been invoked. Exiting Full Screen mode will return the project window to the original Space that it came from.

[3] Apple's term for virtual desktops, where you can create a fresh empty desktop for a particular session of work.

The menu command by this name was previously used to invoke Scrivener's full screen text editing mode. The name of this feature has been changed so as to not conflict with Lion's standard naming convention.

Enter (Exit) Composition Mode (⌥⌘-F) Enters the full screen writing environment, switching to text editing mode if necessary. If the editor is displaying a Scrivenings session, the entire session will be opened in full screen. Note this option is not available to media and web pages, or any other view modes.

This feature used to be named "Full Screen". It has been changed to avoid conflicts with the Lion feature by the same name. Even if you are not running Lion, the new name will be in use for the sake of consistency.

Composition Backdrop ► Choose a background image for your full screen environment, rather than just using the plain background colour. The image will be stretched to fit the screen vertically. For your convenience, all of the image resources in the project will be listed in this menu, or you can select **Choose...** to select an image from your computer. To remove an image you've already set, select **None**.

Show (Hide) Toolbar Toggles the visibility of the main application toolbar.

Customize Toolbar... Lets you configure the toolbar of the main window so that it contains the commands that you most use in the order most intuitive to you. This can also be done by right clicking on the toolbar itself.

A.5 Project Menu

The Project menu addresses commands and configuration options which are specific to the active project, such as document format over-rides, custom meta-data, statistics, and more. If more than one project is open, this menu will target the foremost project window.

New Text (⌘-N) Creates a new text document below the current selection. When a folder is selected, the new file will be created within the folder. Otherwise the new file will appear as a sibling below the current selection, or the location of the document that is being edited.

When the current position for a new item is within a container that has a default subdocument type (subsection 8.5.5) assigned to it, the menu label will change to reflect that. So for example if the current folder has a default subdocument type of

"Interview Transcription", then the menu label will read, "New Interview Transcription".

New Folder (⌥⌘-N) Creates a new folder document at the current selection. Folders will always be created as a sibling to the current selection.

For more information on the differences between files and folders, see Project Planning (chapter 12).

New From Template ► Once a template folder is set, this menu will also be populated with the children of that template folder, which may need to be set with **Project ► Set Selection As Templates Folder**, below. Read more about Document Templates (section 8.5).

New Media File ► This sub-menu accesses the audio and camera recording features of your Mac, if applicable. See Recording Audio and Photo Notes (section 21.4).

For tricks on how best to use text and project statistics, read Goals and Statistics Tracking Tools (section 21.3).

Show (Hide) Project Targets (⇧⌘-T) Toggles visibility of the targets floating panel, which lets you track a few simple metrics in real-time as you type and edit. This can be left open while you work.

Project Statistics (⇧⌥⌘-S) Opens a window displaying the statistics for the current project, including the word and character count for the draft and for the current binder selection.

Text Statistics (^⌥⌘-S) Opens a window displaying statistics for the current document (and thus only available when a text document or Scrivenings are focussed).

Project Notes (⌥⌘-P) Displays the project notebook in a separate window. For more information on using project notes, see Project Notes (section 12.3).

Show (Hide) Project Keywords (⇧⌥⌘-H) Displays or closes the Project Keywords (section 10.3), for managing all of the keywords in use by your project or assigning them to items.

This feature has been renamed from "Keywords HUD".

Meta-Data Settings... (⌥⌘-,) Displays a sheet which lets you adjust the two main document meta-data fields (Label and Status by default, but these can be renamed), custom meta-data as well as the general project information (such as author and title) for this project.

See also: Setting Up Meta-Data (chapter 10).

Auto-Complete List... Displays a sheet which lets you manually add and remove terms to the project's auto-complete list. Depending on your auto-completion settings, this will let you add custom names, places, and workflow indicators to assist in typing out repetitive items. Modify how aggressive auto-complete should behave in the Corrections preference tab (section B.9).

Text Preferences... Displays a sheet which presents a configuration pane very similar to the one found in the Formatting tab (section B.8) of application preferences. Changes made to this panel will override the application level settings. It works in the same way as the application defaults, affecting all *new* documents, but not existing ones. Additionally the linked footnote mode can be modified to use an anchor marking, instead of highlight words and phrases in the main text editor. Read more about Setting Default Project Formatting (section 12.4).

Set Selection as Templates Folder / Clear Templates Folder Will convert the selected binder item into a special template folder (though note it could also be a document stack). Each project can only have one template folder at a time. If more than one item is selected in the binder, this menu item will be disabled.

When a Templates folder has already been set in the project, this menu item will changed to "Clear Templates Folder", removing its designation, but not it or its contents.

Empty Trash... Permanently discards the current contents of the project trash can. You will be warned before this is done. Once contents have been deleted, there is no way to undo that action. This can also be done by right-clicking on the Trash itself in the binder.

A.6 Documents Menu

The Documents menu contains all commands relating directly to documents, such as duplicating and splitting documents, viewing them in combination, moving them, and hoisting the binder to view isolated parts of the draft.

Open ► The Open sub-menu provides you with a number of ways in which to open a selected item. These menu items are relevant from the binder or collection views, as well as corkboard and outliner views, but note that in most cases, a single click, or double-click on the icon will open a file all by itself. How it does that is dictated by the settings available in the View ► Binder Affects... sub-menu. The corkboard and outliner can be configured to immediately open clicked files too.

The first two menu items open the selected documents in either the current editor or the other split. To open an item in the current split (and replace the existing view), use ⌥⌘-O. To open selected item(s) in the other split, use ⇧⌘-O. This will open a new split if necessary, using the previous split orientation. The actual labels of these menu items will change, depending upon the current editor that is active, and whether or not split orientation is horizontal or vertical. For example, if the focus is in the bottom editor of a horizontal split, the label for ⌥⌘-O will be "in Bottom Editor".

as QuickReference Opens the selected document in a QuickReference panel, rather than any active or inactive splits. This also works for the current text you are editing. When in a Scrivenings session, only the current document out of the session will be opened as a QuickReference panel.

in Quick Look This menu command is only available for types of files which Scrivener does not recognise. It will hook into the Mac's Quick Look system and display that file according to its Quick Look abilities.

(With) Compilable Subdocuments Working in a manner similar to "With All Subdocuments as Flat List" (below), this creates a view with only those items which have been set to Include in Compile (subsection 10.1.5). Unlike the previous tool, this command uses your current view mode, so this will produce a Scrivenings session, corkboard, or outline view depending on your preference, though if the result of the action produces only one text document, it will revert to using the text editor.

Read more about Filtering Items in Views (section 9.9).

With All Subdocuments as Flat List ► If the selected item has subdocuments (this can be determined on the corkboard if the index card appears as a stack of cards; and in the outliner if it has an arrow beside its name) you can choose to open it as a corkboard either in the current editor or the other split. This command is essentially a macro for selecting the container and all of its children at once, producing a flat corkboard in Multiple Selection mode.

- *On Editor Corkboard*: (⇧⌥⌘-O) Replaces the current editor view with the selected item's contents. This is similar to pressing ⇧⌘-O to open selected cards on the corkboard, only it also will add all of their descendants to the corkboard as well.
- *On Other Editor Corkboard*: As with above, but uses the split editor to open the item, instead of replacing the current view, creating a split if necessary.

---

in External Editor (^⌘-O) This command is available on any type of item that is not a text or folder document. It will open the file using the default editor for that type of file. An example would be opening a picture in Preview. Opening items this way allows them to be edited in their native applications, and the results of those edits (if Scrivener can display the file type) will be visible upon saving that file.

Snapshots ► Provides commands to manage a document's snapshots, make new ones, and commit large-scale snapshot actions on selected items. For more information on how to use snapshots, see Using Snapshots (section 15.6). Note that in some cases these shortcuts will not be available. Generally this happens when the selection includes items which are not text items, or the text of the item(s) are empty.

Take Snapshot (⌘-5) Available when viewing any text or folder document, this command will take a snapshot of the current text and store it for later use.

When more than one item is selected, the title of this item will change to "Take Snapshots of Selected Documents". All selected items will be snapshot.

Take Snapshot With Title (⇧⌘-5) As above, but will prompt you for a descriptive title prior to taking the snapshot, which will be displayed next to it for future reference.

When more than one item is selected, the title of this item will change to "Take Titled Snapshots of Documents". The supplied title will be applied to each snapshot taken for all selected items.

Show Snapshots Will reveal the Snapshots pane in the inspector, revealing it if necessary. While this menu item does not have a shortcut, an identical function in View ▶ Inspect ▶ Snapshots provides the shortcut key, ^⌥⌘–M.

Show Changes ▶ When a snapshot is selected from the snapshot list, it is possible to view the changes between that snapshot and the current document. Note these features are also accessible from the snapshot pane itself, in a button at the top of the inspector. If two snapshots are selected, these two snapshots will be compared between each other, using the oldest snapshot as the exemplar. This can also be done in the main editor by dragging a snapshot from the list to an editor header bar while holding down the Option key.

- *Compare/Original*: Toggles visibility of the change tracking mode.
- *Comparison Granularity*: These options can be toggled on and off individually to determine how closely changes will be tracked, with the finest level selected being the used method. *By Paragraph* will only mark changes at the paragraph level; *By Clause* only marks changes made at the phrase level; *By Word* is the most detailed level of change tracking available. Use these settings to fine-tune the results if what you are getting is too precise or vague to be useful.
- *Next Change*: (^⌘–)) Scrolls the view to the next available difference in texts.
- *Previous Change*: (^⌘–() Scrolls the view to the previous difference.

The following two commands will only activate when a snapshot selection is active in the Snapshot Inspector pane.

Roll Back Rolls back to the selected snapshot, replacing the text in the main editor. The option to snapshot the *current* editor text, before making this revision, will be given in a pop-up dialogue box.

Delete Permanently removes the selected snapshot from the disk. You will be asked to confirm this action as it cannot be undone. You may also delete snapshots with the backspace key, or by clicking the - button in the Snapshots Inspector.

Duplicate ▶ There are two different methods you can use to duplicate the selected item(s).

- *with Subdocuments and Unique Title*: (⌘-D) This is the default method in most applications. The entire item (including all of its children if it has any) will be duplicated, and it will be provided with a unique name automatically.
- *without Subdocuments*: (⇧⌘-D) Also referred to as "Simple Duplicate" in prior versions of Scrivener, this is most useful when you only want to duplicate the container of an item by itself, such as a chapter folder, but not all of the scenes within it. In this case, a unique name will *not* be provided.

Set Selection as Title (⇧⌥⌘-T) Available when text has been selected in the main editor, the title for the current document will be replaced with whatever text has been selected.

Move to Trash (⌘-Delete) As you no doubt guessed, this moves selected items to the project's trash. It will use active selections in the Outliner or Corkboard, as well as in the binder sidebar.

Split ► Provides a couple of methods for splitting the current document into two different documents. In both cases, everything below the current cursor position will be moved into a new document below the current one. Everything above the cursor will remain unchanged. This action cannot be undone (except via the Merge command, below).

- *at Selection*: (⌘-K) Simply create a new document with all text below the current caret position. In cases where text has been selected, the caret position will be considered as the start of the selection range.
- *with Selection as Title*: (⌥⌘-K) Works the same as above, but in this case the currently selected text will be used to title the new document that is created, rather than requesting a new document name after pressing the shortcut.

Merge (⇧⌘-M) The opposite of splitting, merge will take two or more documents and combine them all together into a single document, using the top-most document in the selection as the "template" for any of the merged document's meta-data that cannot be otherwise combined, such as labels and status. This action cannot be undone (except via the split command). In those fields where combination makes sense, such as notes and keywords, all of the documents will be used to create a combined meta-data result.

How documents get merged together can be controlled in the General preferences tab (section B.2), under Separators.

Group (⌥⌘-G) Takes one or more selected items and creates a new folder, placing the selection within that folder. You will be given the option to name that folder after invoking the command.

Ungroup (⌥⌘-U) The opposite of grouping, ungroup will move the contents of the selected container up one level so that they become siblings of that container. This action will not destroy the original container, allowing you to move the items back in, if it was made in error.

Move ► Provides keyboard shortcuts for moving the item around in the outline in two dimensions, as well as completely displacing it to another spot in the binder.

- *To*: This menu will be populated with all of the items in the binder. The selected item(s) will be moved beneath the selected item as children. Note that if you select a document as the target, it will become a document stack as it now contains children. Items which have been used recently will be displayed in the top-level of this list, beneath any favourites (if applicable).
- *Left/Right/Up/Down*: Use ⌘-^ with the appropriate keyboard arrow key to move items around in the outline. Left and Right will promote or demote the item in the hierarchy; Up and Down will change its order in context with its siblings. On the Corkboard, which does not show hierarchy, the arrows will move cards around spatially amongst one another.

Add to Collection Displays a list of all standard collections in the project. The currently selected item(s) will all be added to the collection you choose. This command is also available in the binder contextual menu.

- *New Collection*: in a single move will create a new collection, opening the tab interface if necessary, and assign the currently selected document(s) to that new collection. This is equivalent to clicking the + button in the collection header bar, though more convenient if you use the menu and realise while there that what you thought was a collection has not yet been actually made.
- The remainder of this menu will be used to display existing arbitrary collections. Dynamic collections will not be displayed as they cannot be modified directly.

Sort ► Sort acts on the children of a selected container (in the Binder or Outliner), or on the contents of the current Corkboard, if that corkboard is displaying a single folder (not a multiple selection).

Items will be alphanumerically, either in *Ascending* or *Descending* order. Note that this a **permanent** change that cannot be undone. Additionally, you can choose to shuffle all of the children randomly.

Hoist Binder Mimics the behaviour of traditional outlining software, allowing you to temporarily obscure the full binder and focus on only one portion of it. In other words, hoisting will display the selected container all by itself in the binder sidebar. Unlike collections, hoisting the binder retains *all* of the binder's abilities. Changes made to ordering and structure while hoisting will be made to the book outline itself. If you want a non-destructive way of viewing portions of the binder, it is recommended you use Collections (section 8.4) instead.

Unhoist Binder Returns the binder to full display.

Favorites ► Provides commands to manage your favourite documents. "Favorites" will be displayed in various places around the interface, particularly in item selection menus, at the top of the list, such as "Go To", "QuickReference", "Scrivener Link", and more. Read more about Using Favourites (section 9.6).

Add to Favorites Adds the selected item(s) to the favorite list. Can be used from either the text editor (the document you are currently editing will be the target, in Scrivenings mode) or with a selection of one or more items in the Binder or group view modes.

Remove from Favorites Will remove the selected item(s) from the favorite list, ignoring those selections which are not already there. As with adding favourites, this command can be used from the editor or group views.

Clear Favorites Wipe out the entire favourite list with one command.

Manage Favorites... Displays the floating Favorite Management window wherein you can change the order in which they will appear in the interface, remove one or more selected items, or drag new items into the palette to add them to the list.

Convert ► Provides tools for converting the document's type, writing mode, or format defaults.

- *to Folder/File*: Converts the selected item(s) to folders or files. Note that these two are very similar in Scrivener, so this tool makes it easy to change your mind later about whether or not something should be a folder or a file.
- *Web Page to Text*: Activates when viewing a WebArchive item. Will analyse the archived page for its text content and create a rich text alternative that is editable and considered a text document from that point onward. Note this cannot be undone, create a duplicate of the web archive, first, if you wish to retain a copy with its full layout.
- *Formatting to Default Text Style*: If the font and ruler styles of the document do not conform to the application or project defaults, you can use this command to use those to reset its appearance. Read more about Resetting Formatting (subsection 15.4.5).
- *Script Format...*: Useful when you need to switch from one scripting standard to another. Note that if all you wish to do is change the current document writing mode, you can press ⌘-8 to do so. This function is strictly for converting between pre-existing scripts to bring them up to spec. For more information on scriptwriting, see Scriptwriting (chapter 20).

Auto-Generate Synopsis For single items, you can click the button in the upper right-hand side of the index card in the Inspector. When you wish to perform this automated function on many items at once, use this menu command. The command will work in the same fashion as imported text documents, taking the first 500 characters of text in the document. Note this cannot be undone; do not run this command on cards you've already manually entered synopses for, unless you are sure you want to replace them. Scrivener will warn you if it looks like you are about to overwrite custom synopses.

Append Synopsis to Main Text Operates in an inverse fashion from above. The contents of the index card's synopsis field will be appended to the end of the main text editor, corresponding to the selected card(s). Useful if you've done a little pre-composition in Outliner or Corkboard mode, and now wish to push those ideas into the manuscript text.

This command will be inactive unless the selection contains at least one card with a synopsis. Selected cards without a synopsis will be ignored.

Default New Subdocument Type Only available when document templates are in use for the project. This will set the default template item to use instead of the basic vanilla text file, when adding child items to the selected container. See Default Subdocument Types (subsection 8.5.5).

Change Icon ► Provides access to existing icons and lets you change the document icon to a custom icon for the selected item(s). Icons can also be managed either globally, or by project, from here. Your custom icons will be available, as well as many built-in icons that you are free to use in any of your works.

- ❧ *Reset Icon to Default*: Removes custom icons from the selected item(s). If no custom icons are in place, this menu item will do nothing.
- ❧ *Manage Icons...*: Reveals the Manage Icons Palette, where you can add or remove your own custom icons either to the project, or make them available globally to all Scrivener projects. For more information on managing and creating icons, see Custom Icons (section 8.6).

A.7 Format Menu

The Format menu contains all commands that deal with formatting the text of an individual document. It contains all of the usual OS X commands (previously found in the Text menu in older versions of Scrivener) for changing the font, setting line spacing, creating tables and lists and controlling the ruler, along with commands for highlighting and setting the current text colour. Additional Scrivener specific tools for formatting the text can be found here as well, including various forms of notation, text conversion utilities, and bibliography software integration.

Font ► Displays the standard OS X Font submenu, which contains useful tools for controlling the character-level appearance of your text. Fonts family, variant, and size can more easily be controlled with the The Format Bar (subsection 15.4.2).

- ❧ *Show Fonts*: (⌘-T) access the system-wide standard font palette, which will change the font you type in from that point forward, or alter the existing font of a selected range of text.
- ❧ *Bold/Italic/Strikethrough*: all of these operate in the same fashion by either modifying the future behaviour of typing if there is no selection, or toggling the selected text between the various styles. Note that not all fonts support these styles. If it appears that one of these is not working, check your font for proper variant support. Strikethrough can also be applied to text in PDF documents.
- ❧ *Underline*: has been expanded from the system default menu item to provide additional support for underlining styles, including dotted, dashed, and by word.

- *Remember Typing Style*: use to set up fonts and ruler styles in blank documents. Ordinarily, if you change formatting settings on a blank document, these settings will be removed to optimise the RTF file. Use this command to force these settings to be retained. Generally you will only need this when setting up empty Document Templates (section 8.5). Since having text in a file preserves any formatting made to that text automatically, this menu command will be unavailable in file that has content in the main text editing area.
- *Outline*: changes the style of the font to outlined instead of filled.
- *Bigger*: (⇧⌘-=) increases the size of the font by one point.
- *Smaller*: (⌘- -) decreases the size of the font by one point.
- *Kern*: supplies access to the OS X typographic kerning engine.
- *Ligature*: if the font family supports typographic ligatures, they can be enabled or disabled here.
- *Baseline*: adjust the baseline height of the text, most commonly referred to as superscripting and subscripting.
- *Character Shape*: if the font family supports typographic shape alternates, they can be adjusted here, such as Old Style Numerals.
- *Show Colors*: (⇧⌘-C) toggles the visibility of the colour selection palette. Note that Scrivener uses this palette in a number of different places, not strictly related to text colour.
- *Copy/Paste Font*: (⌥⌘-C / ⌥⌘-V) copies the font settings from the current cursor position. Use Paste Font to apply these settings to other selections of text.

Text ► Displays the standard OS X Text submenu, which contains useful tools for controlling the paragraph-level formatting of your text. Paragraph alignment and spacing can also be controlled with the The Format Bar (subsection 15.4.2).

- *Align Left/Center/Justify/Right*: controls the text alignment of the current paragraph, or all paragraphs falling within a selection range.
- *Writing Direction*: toggle between Left-to-Right and Right-to-Left writing styles.
- *Keep with Next*: inserts an invisible character which will keep this paragraph together with the next paragraph, so that they will not become separated by a soft page break.

- *Copy/Paste Ruler*: (^⌘-C / ^⌘-V) Copies the ruler and paragraph format settings from the current paragraph. Use Paste Ruler to apply these settings to other selections of text.
- *Tabs and Indents...*: provides a panel which can be used to set indents and tab stops to the ruler with precision units. If you find dragging the icons on the ruler to produce unreliable results, or if you cannot hit a desired unit of measurement precisely, this tool will make it easy to add stops and indents where you need them.
- *Remove All Tab Stops*: A convenience tool to delete all tab stops in the current paragraph or selection.
- *Line and Paragraph Spacing...*: accesses the advanced paragraph and line spacing palette, which can also be found on the Format Bar, off of the line-spacing sub-menu.

Increase/Decrease Indents ► Handy controls for indenting text in a variety of fashions, or to globally increase or decrease the indentation level for the selected paragraphs. All method adjust indentation in increments of 0.5cm; 0.25in; 1.5pc; and 18pt.

- *Increase/Decrease Indents* (^⌥⌘-RightArrow or ^⌥⌘-LeftArrow): moves the indent level for all items inward or outward. This command will maintain the overall indent format of the paragraph. If it is a hanging indent, for example, it will remain a hanging indent, only with both indent settings moved one way or the other.
- *Increase/Decrease First Line Indent*: adjusts the first-line indent marker inward or outward.
- *Increase/Decrease Hanging Indent*: adjusts the indent value independent of the first-line indent, forming a "hanging indent" where the first line is leftward of the rest of the paragraph.
- *Increase/Decrease Right Indent*: adjusts the block indent level, moving both the first-line and main indent markers at an even increment. Useful for block quotes.

Table ► Inserts a default table into your document at the cursor position unless you are already in a table. In both cases it will also open a floating configuration window containing tools for adding and manipulating tabular data in the text editor. Tables can be set up with a given number of rows and columns, but using the options in this window, you can add or remove either to fit the constraints of your data

later on. You can also normalise the width and height of columns and rows respectively with the Distribute functions and sort rows ascending or descending by the alphanumeric quality of the first cell for each row.

The management commands in this menu are also available by right-clicking on a table in your editor. The specific commands are documented here (section 15.3.2).

Lists ► Supplies basic list styles, and few advanced controls for bullet and enumerated list management. The list creation aspects of this menu are also available when clicking on the list button in the Format Bar.

- *Next/Previous List Style*: (⌥⌘–RightArrow / ⌥⌘–LeftArrow) selects the next or previous built-in list style available and converts the current list to that style.
- *Custom List...*: build your own custom list type using the standard OS X tool for doing so (refer to TextEdit help for how to use this panel).

Formatting ► Provides tools for creating and applying ruler and font presets[4] to your text. For more information on using the presets system, including its menu commands, see Formatting Presets (subsection 15.4.3).

Scriptwriting ► Although Scrivener is not a dedicated script writing application such as Final Draft or Montage, the Scriptwriting submenu does make it relatively straightforward to create first drafts which can be later fine-tuned in these applications. For full documentation on this, see Scriptwriting (chapter 20).

Show (Hide) Ruler (⌘–R) Toggles the visibility of the tab-stop and page margin ruler for the active editor. Note that all formatting tools have been moved from this ruler to the format bar.

Show (Hide) Format Bar (⇧⌘–R) Toggles the visibility of the character and paragraph formatting bar, which provides quick access to many of the most common types of formatting tools.

---

[4]Note that Scrivener does not supply word processor stylesheets. presets are strictly saved formats which can be applied to your text. Once applied they are static changes to the text and will not be updated later if you change that favourite style.

Highlight ► The highlight menu lets you place a background highlight behind the selected text, much like using a highlighter marker on paper. In addition to the five provided presets, this menu also displays any favourite colours that have been set on your system. For more information on how to use highlights, see Text Colour and Highlights (section 18.4). Highlighting (in basic yellow) can also be applied to text in PDF files.

Revision Mode ► Revision modes force anything you type to be displayed in a provided colour until you disable the revision mode. This menu provides access to these levels. For more information on using this, see Marking Revisions (section 18.5). The first menu slot will alternate depending on whether or not you have any selected text in the active editor. The two entries will be marked accordingly below.

Remove Current Revision Color (No Selection) Only active when one of the revision levels has a checkmark beside it. This will strip the active colour from the active editing session. As with many commands that work on this level, it works across an entire Scrivenings session if applicable.

Mark Revised (Selection) When a text selection has been made *and* a revision level is in use, this will let you mark text ranges with the current revision colour. This can also be done by clicking on the text colour tool in the Format Bar, as it it will be reset to the current revision level while in use. Select the text you wish to mark, and click the text colour chip, or use this menu command.

Remove All Revisions Indiscriminately removes all five revision level colours from the active editing session *or* text selection.

For more information on annotating and footnoting your text, see Annotations and Footnotes (chapter 18). Note that while the menu system refers to these as footnotes, it is possible to export them as endnotes during compile.

Comment (⇧⌘–8) Attaches a comment to the selected text (or previous word), making a new linked comment in the inspector pane. Useful for keeping notation text out of the editor.

Footnote (^⌘–8) As with Comments, adds a linked footnote style comment to the inspector pane. Useful for keeping footnote information out of the editor.

Inline Annotation (⇧⌘–A) Converts the selected text into an inline annotation, or toggles the writing mode to annotation mode. Useful for placing notes right in the editor, alongside the relevant text.

Inline Footnote (⇧⌘–F) Converts the selected text into a footnote, or toggles the writing mode to footnote mode. Useful for keeping footnotes directly in the editor session.

Convert ► The convert submenu provides useful tools for the manipulation of text and notation features.

To Uppercase Converts the selected text to all uppercase characters.

To Lowercase Converts the selected text to all lowercase characters.

To Title Case Converts the selected text to title case, which takes into account common English grammar conventions.

To Small Caps If the current font supports proper small caps, the text will be converted to that variant. However it will fake the results using font sizes otherwise. If your workflow requires true small caps, be certain the font you are using can handle them.

Quotes to Smart/Straight Quotes Converts between typographic "curly" quotes and straight inch and foot marks.

Multiples Spaces to Space Useful for cleaning up a document that has been typed up with multiple spaces in between sentences and so forth.

Strip Leading Tabs If the original document was typed in the fashion that one might use a typewriter, with tab characters in front of every paragraph, this command can be used to strip them all out.

Bold and Italics to MultiMarkdown Syntax Converts rich text bold and italic formatting codes into respective MMD syntax. Note that in some cases this can produce odd results; be sure to proof the results before sending off to the printer.

Inline to Inspector and Inspector to Inline Conversions These four commands allow you to convert inline annotations to linked Inspector comments, vice versa, and footnotes. These commands work on the entire current document.

Options ► Provides options which impact the display or behaviour or the text editor.

- *Show Titles in Scrivenings*: when enabled, editable titles of documents will be placed in the editing session along with their associated texts. Titles can be independently styled to set them apart in the Formatting tab (section B.8). These will only appear in Scrivenings mode.
- *Show (Hide) Invisibles*: toggles visibility of hidden control characters, such as paragraph breaks, table cells (when otherwise set to have no borders) tabs, spaces and page breaks.
- *Show Line Numbers*: adds a paragraph (line feeds within a paragraph will not be counted) numbering feature to the standard editor (this feature does not work in Page View mode). Paragraphs will be numbered in the left column. You can also optionally elect to have Scrivener only count every fifth paragraph.
- *Show (Hide) Compile Footnote Numbers in Inspector*: On by default, after you compile and all footnotes have been counted, their numbering will be printed in the Inspector alongside the footnote itself (section 18.2.3). You can also optionally instruct the compiler to ask you if these numbers should be updated, every time you compile.
- *Typewriter Scrolling*: (^⌘–T) toggle a feature which keeps the currently edited line of text in the middle of the screen, vertically centred. This setting is global, but will impact composition or the regular editors independently, depending upon the current mode.

Bibliography/Citations... (⌘–Y) If a citation manager has been set up in the General tab (section B.2) of Scrivener's options, this command will launch or activate the set manager, facilitating the process of selecting a citation and paste it back into Scrivener using whatever system that application provides.

A.8 Window Menu

The Window menu contains commands for controlling windows and panels along with a list of all open windows.

Minimize (⌘–M) Minimises the window to the Dock (the same as the yellow button in the top-left of the window). Hold down the Option key to minimise all.

Zoom (^⌘– -) Zooms the window in or out (the same as the green button in the top-left of the window). Scrivener will try to zoom intelligently to best fit the contents

of the window, taking into account your preferred editor width. Hold down the Option key to change to Zoom All.

Zoom to Fit Screen (^⌘-=) Zooms the window in and out, so that it takes up the whole of the screen.

Layouts / This Sub-menu is populated by the saved layouts you've created with the Layouts panel. It allows easy access to switching these sets of view configurations.

This menu will not appear if no layouts have been saved on your computer.

Float Window "Floats" the project window above other application windows, so that you can have it visible at all times should you wish, even when switching to other programs. Be sure not to use this in conjunction with Zoom to Fit Screen.

Float QuickReference Panels (^⌘-Q) Toggles whether or not QuickReference panels should float over the project windows. This setting impacts all QuickReference panels created from the current project. Panels from another open project will behave in accordance with that project's settings.

Show (Hide) Scratch Pad (⇧⌘-Return) Displays or closes the floating Scratch Pad Panel (section 11.3); useful for collecting information while using other applications.

Bring All to Front Brings all Scrivener windows to the front. Hold down the Option key to change to Arrange in Front.

Closed QuickReference Panels During a single session, any QuickReference panels that have been closed will be saved into this sub-menu, giving you quick access to them if you need to re-open them.

The rest of this menu will be populated by all of the windows (excluding panels, like the Project Keywords window) that are currently open in Scrivener. Selecting from this list will bring that window to the front. QuickReference panels will be separated into their own section, if any are open.

A.9 Help Menu

The Help menu provides standard access to your Mac's menu search utility (available in 10.6+), as well as useful tools and links for learning Scrivener, or getting in touch with us.

A note on performance: large projects that contain many items in the binder will cause this menu to exhibit a noticeable delay, thanks to the menu search feature. This is a limitation in Apple's code, and there is no good way around it.

The Search menu is the only menu that can be directly activated from the keyboard. Tap ⇧⌘-? to call it up. This will place the cursor in the search field (on 10.6+), so you can begin searching immediately.

Scrivener Manual A quick link to the PDF that you are likely reading. The version that ships with Scrivener will be kept as up to date as possible, but newer revisions might also be available on the web site's support page[5].

Interactive Tutorial If you have not yet gone through the tutorial (you should!) this menu command will walk you through the process of creating the tutorial Scrivener project. You will be given the option of where to save it, and from that point on you can load it like an ordinary project. If you have already created the tutorial project, you can use this menu command to quickly load it again, so long as it hasn't been deleted or moved.

Video Tutorials A handy hyperlink to our web site's video page[6]

Placeholder Tags List... There are many placeholder tags you can use with Scrivener, and this will provide an exhaustive list of them all in a floating window so you can copy and paste them into your project (or compile settings) as needed.

Support A handy hyperlink to our web site's support page. Here you can download the latest copy of the PDF in US Letter or A4 (or the project used to make it), find contact e-mail addresses, links to our forums and wiki, after-sales support from our vendor, eSellerate, and more.

User Forums A handy hyperlink to the official Scrivener forums where you can meet other authors around the world using Scrivener, share tips, report bugs, request tech support, or have a cup of latte in our off topic section.

Release Notes Hyperlink to the official release notes[7] web page on the web site.

[5] http://www.literatureandlatte.com/support.php
[6] http://www.literatureandlatte.com/videos.php
[7] http://www.literatureandlatte.com/scrivChangeList.php

Literature & Latte Home A handy hyperlink to our home page[8] which provides easy access to everything else we offer on our web site.

Scrivener Home Link to the main Scrivener page[9], where you will find useful download links for updates, case studies, links to share Scrivener with Twitter and Facebook, and more.

Keep Up to Date... Presents a form which you can use to submit your e-mail address and name to subscribe to our newsletter. This is a low volume list that we use to send out important updates and news. Please take care to whitelist "litereatureandlatte.com" in your spam filter, prior to submitting this form, as you will be sent a confirmation e-mail which must be responded to before you will be subscribed. You can also click on the Twitter and Facebook buttons to visit our official social pages.

Privacy Notice: Your e-mail address will never be shared with third-parties or sold to marketing lists.

[8] http://www.literatureandlatte.com/
[9] http://www.literatureandlatte.com/scrivener.php

Appendix B

Preferences

Like most applications, Scrivener installs in a "ready to use" state. The default preferences have all been carefully selected to present a cohesive and useful writing environment. Preferences should thus be considered an optional, and perhaps even slightly more advanced way of using Scrivener. This manual has described the application in reference to its defaults, but there are a great many ways to tweak the interface and behaviour of the application to suit your preferred working style. This appendix will thoroughly document every single preference item, as presented to you in the Preferences window itself. This way, if you wish to seek clarification on a particular option, you can find it using the manual's list of preference tabs, and then within that section, sub-sections will separate groups of preferences in accordance with their visual grouping in the application.

Because Scrivener is a multi-project application, preferences are sometimes split into different areas so that you can supply new projects with some starting preferences, but allow some of them to deviate from the norm, if needed. Once these projects have been created, they will maintain their own individual settings from that point forward. One step beyond that, there are a sub-set of preferences which apply at the document level, and each document can diverge from the project and application preferences in certain ways.

Not all preferences work in this "cascading" fashion. The ones that do will be indicated as such in their notes. Many of the preferences that are wholly project specific make no appearance in the application preference system at all. A good example is whether or not labels are displayed on index cards. By default they are, but you can turn them off for a project and it will remember that setting. There is, however, no way to change that behaviour globally[1].

All of the project specific preferences are located in menus or in the Footer Bar for the respective view, and of those in menus, most are in either the Project menu or the View menu. This section will not cover these specifically, though some might be mentioned in

[1] Actually, initial project settings are handled by the template you choose when creating a project, so you should consider project templates a form of "secondary" preference. Read more on Templates (section 7.10) to learn how to use these to adjust Scrivener's behaviour with a brand new project.

passing where applicable. Instead, this section strictly focusses on the application preferences dialogue. For information on specific menu options, see the appendix Menus and Keyboard Shortcuts (Appendix A).

In the preference tab, you will be presented with row of icons along the top. Each of these icons leads to a sub-section, or tab, of the application preferences. The rest of this appendix will deal with each of these sub-sections in turn. If you close and re-open the preferences window later in a single session, the last sub-section you were looking at will be preserved.

As with many Mac applications, preferences are enacted and saved as soon as you change anything. This means you do not have to worry about clicking any buttons once you are done; just close the preferences window. It also means that if your screen is large enough, you can leave the preferences open to the side and tweak things until you get them looking and acting the way you want.

Along the bottom of the preference window you'll find a **Manage...** drop-down menu and a button labelled **Defaults**. The latter button will do a "factory reset", returning everything in the preferences to their default states. This cannot be undone, but you will be warned after clicking the button and given a chance to cancel.

B.1 Preference Presets and Themes

The **Manage...** drop-down menu has several options available for organising or backing up preferences, as well as access to any presets you might have saved. This can be particularly useful if you work on more than one computer, and in the case of themes, if you prefer to use darker colours at night and brighter colours in the day.

Load/Save All Preferences Save your preferences to an external file for either backup purposes, or to synchronise your settings between multiple computers. Preferences can be loaded, replacing the current settings *completely*, with the Load command.

Save All Preferences as Preset... Presets are simply a convenience feature for organising different preference sets. In cases where you need to use varied settings for different projects, this will make it easy to switch sets on the fly. The name that you provide will be displayed in this menu in the future.

If you wish to update an existing preset, make any changes you see fit, and then use this command to save the preset under the same name. You will be asked for confirmation before the old preset file is overwritten.

Load/Save Theme Preferences Themes differ from the above in that they only save and load partial preferences. They only operate on those preferences which govern appearance, such as interface colours (standard and composition mode), corkboard

and outliner appearance, and optionally, fonts. Fonts can be saved into a theme by checking off the appropriate boxes in the Save Theme window. These commands work with files, and so are useful when synchronising settings between computers, sharing your theme settings with others, or loading theme settings you've downloaded from the Internet.

Save Theme Preferences as Preset... Similar in function to saving all preferences as a preset, this makes it easier to store and switch between multiple theme settings without having to manage individual files.

B.2 General

General preferences govern the application's basic behaviour, its integration with other programs, and how appending or merging text with existing documents should be presented.

B.2.1 Startup Options

Reopen projects that were open on quit Whenever Scrivener is launched, projects that were not closed when you last quit will be automatically reopened, bringing you straight to what you were last working on.

Show template chooser when there are no projects open Toggles the display of the start panel, which lets you create new projects, open existing projects on your drive, or select from a list of recently worked on projects.

Reopen QuickReference panels when opening projects Ordinarily, QuickReference panels (section 21.2) are session based. If you close the project and come back later, they will all be gone. With this preference enabled, all projects will remember any QuickReference panels left open when it was closed, and will re-open them for you the next time you load the project.

Language By default, Scrivener will attempt to select the appropriate interface language based upon your system settings. If this does not work, or you would like to use another interface language for Scrivener, you can use this setting to manually adjust the language.

Use the "Default" selection to return the interface to the Mac system default.

Automatically check for updates If this is checked, whenever Scrivener is launched it will check the website to see if there is a newer version available and, if so, will ask you if you want to update. Note that if you do not have this checked, you

can still check for updates by choosing Check for Updates... from the application (Scrivener) menu.

The drop-down menu beside this option governs how frequently Scrivener will check for new versions. If you want to stay on top of the latest updates, "Daily" is generally good enough, but if updating frequently is disruptive to your work habits, you can set this down to check by week or month.

B.2.2 Saving

Auto-save after *n* second(s) of inactivity Scrivener automatically saves changes made to projects, but so that it is not constantly saving—and to avoid any slowdown and interruption to your work this might cause—Scrivener waits to save until you stop interacting with the program for more than two seconds (by default). You can change the period Scrivener waits to auto-save with this preference. If you increase the period significantly, be sure to use File ▸ Save regularly to force saves to your project. Scrivener will always auto-save when you close a project.

Automatically name untitled text documents upon saving When enabled, text documents that have been left untitled will acquire the first few words from the document. This will not trigger until you have added a second line or typed around one complete sentence into the first line. This way, your documents do not get titled until the wording has settled down a bit.

Take snapshots upon manually saving text documents Typically, taking snapshots is up to you. If you would prefer automatically created snapshots whenever you force save the project with File ▸ Save..., this option will snapshot all text items, and only those, that have had their main text content altered during the session or last time you saved. Snapshots that have been automatically generated in this way will have their name set to "Untitled (Save)" to differentiate them from normal "Untitled" snapshots you take manually.

B.2.3 Services

Options for how Scrivener's clipping services should work. For more information on available services, read Scrivener Services (section 11.2). For the least amount of intervention, set both of these options to off.

Bring Scrivener to front when using Services Choose between active and passive clipping. When this is checked, clipping from other applications will be "active" in that it brings you to Scrivener after you've used a Service. When unchecked, it will

work "passively", staying in the background and allowing you to work in other programs while you add material to Scrivener.

Show title prompt when using clipping Services Normally, when you use a service that requires the creation of a new item in the binder, you will be asked what you wish to call it. When unchecked, this option will defer that task until later, when you are ready to think about names.

B.2.4 Separators

This table lets you configure how Scrivener will combine individual texts when certain actions bring them together. In all cases, you have three choices of divider available:

1. *Single Return*: Inserts a single paragraph break between selections. If two paragraphs appear together, there will be no whitespace between them, save for any style-based paragraph spacing in use.

2. *Empty Line*: A full empty space will be inserted between selections. If you prefer or require working with double-spaced paragraphs, this is the option you will want to use.

3. *Custom*: Lets you type in a custom separator. If you wish to insert carriage returns along with the custom separator, use ⌥Return to do so.

You can assign separators to four different instances:

1. *Merged documents*: Determines how documents will be merged when using the Documents ▸ Merge command. Note that this setting only affects the main text. The inspector notes of merged documents will always be separated by a double newline.

2. *Append clippings service*: When using the global Append Clipping Scrivener service, clippings will be separated from any existing text by this option.

3. *Append selection*: When selecting text and using the Edit ▸ Append Selection to Document command, the selection will be separated from the existing text by this option.

4. *Scratch pad notes*: When appending text to documents from the scratch pad, incoming text will be separated from existing text by this option.

B.2.5 Scratch Pad

Hot key Sets the global hot key for showing or hiding the Scratch Pad Panel (section 11.3). Since this key will be accessible from any application on your Mac, you might need to change this if it is conflicting with another application you use. This tool will warn you if you select a shortcut that is already registered globally on your Mac.

Notes location Your scratch pad notes are stored on the hard drive using RTFD files. This option lets you choose where those files will be automatically saved. By default, they will be stored in your Documents folder, under a special folder that Scrivener will create for them.

In secured working environments, you might wish to alter the location of this to reside in an encrypted drive or disk image on your system. If you do use an encrypted disk image, add it to your login start up list so that it will always be available.

Default format The Scratch Pad utility will always read files if they are saved in one of three different formats: RTFD, RTF an TXT. Files placed into the scratch pad folder manually, will be editable in the Scratch Pad window and remain that format. This setting will only change new notes created from within the window. For the plain-text format, you can supply a custom file extension in the **ext** field. This will mainly be useful if you intend to use some form of markup in your notes. Some external text editors will recognise standard extensions like '.md' for Markdown, and highlight the markup accordingly.

If you use Scrivener for both Windows and the Mac, and intend to keep your notes synchronised between versions using a shared folder, do not use the default RTFD option on the Mac, as this format cannot be read on other platforms.

B.2.6 Miscellaneous

Bibliography/Citations Manager Scrivener can integrate with several popular bibliography management programs. Use this option to select your favourite from the Applications folder. The reset button will clear your choice.

> **Pro Tip**: This merely links which application will launch to the ⌘-Y shortcut. If you do not use a citation manager, but frequently use another program along with Scrivener, you could use this feature to quickly launch or switch to that program.

Warnings When you first start using Scrivener, actions which have a destructive or unusual nature (such as importing in such a way that original formatting might be

lost) will produce a warning dialogue. You can often choose to disable these as you see them from within the warnings themselves. The Reset All Warnings button will clear all of these dismissals and make them appear again the next time it is appropriate.

B.3 Appearance

Most preferences relating to Scrivener's appearance can be modified here. In a few cases, options which modify appearance, but are more related to the specific function they fall within, will be located in the special section for that function. A good example are the colour settings for composition. Other than this option, all interface colour settings are in the Appearance pane. The appearance theme for index cards is set in the Corkboard tab, for another example.

B.3.1 Binder

Show subdocument counts Beside each container in the binder, a number will be displayed showing how many documents it contains. This number is recursive, meaning that it will not only count the container's immediate children but any descendants beneath those children as well.

B.3.2 Outliner

Options for adjusting the appearance of the outliner view mode. Most settings that impact the outliner are project specific and can be accessed either in the outliner footer bar or the View menu.

Has horizontal grid lines Draws separators between rows in the outliner.

Has vertical grid lines Draws separators between columns in the outliner.

Uses alternating row colors Draws alternating background colours behind rows in the outliner. When this is turned off, the background will be a solid shade. You can define the actual colours used in the section below.

B.3.3 Notepad Lines

In document notes Turns on notepad-style ruling for document notes, giving it the appearance of a pad of paper

In project notes Use rule lines in the project notes pane and the project notes window.

Keep text on lines in index cards and notes When enabled, Scrivener will adjust the ruling beneath the text to match the current font size. In most cases, this will produce the most pleasant result, but if your notes have a lot of different font sizes, it might look nicer to keep the rules uniform, regardless of font size. Line rules can also be completely disabled with the above options. Lines in index cards are changed in the Corkboard preference tab (section B.4).

B.3.4 General

QuickReference panels use black translucent "HUD" style Switches QuickReference panels (section 21.2) to using the original black HUD[2] style of window. This is an aesthetic choice which makes little difference on window functionality. You will need to reload your project for this change to make an affect on QuickReference panels that have already been opened once during the session.

Smooth text and line art in PDF documents Anti-aliasing is used by default to keep your PDF documents smoothly rendered at all levels of magnification. If your computer runs slowly while viewing PDFs, disable this to increase performance.

Target progress bars use smooth transition between colors By default, the three colours you use for progress bar display (in the editor footer bar and Project Targets window) will be gradually blended as you type. When this feature is disabled, the progress bar will "snap" from one colour to the next at the 50% and 100% marks. If you prefer a more noticeable indication of when you've reached a goal, this can be a useful option.

Opacity of label colors when used in backgrounds Define how much blending Scrivener should use between label colour and the underlying background colour. Moving the slider right toward "High" will cause labels to be coloured more vividly; moving the slider left toward "Low" will cause the effect to be more subtle.

[2] "Heads-up Display" is an avionic acronym originally used in a military context. HUDs were first employed in the cockpits of fighter pilots, giving them the ability to see their attitude, altitude, speed, radar, and other important flight details without removing their eyes from the horizon and looking down at a console. HUDs have been employed in civilian cases since then. Some luxury cars have HUD style speedometers and tachometers on the windshield so you can stay focussed on the road. In computer usage (and more specifically, Mac software), the term has come to mean any sort of window that provides features or information to the user in a utility role, displayed in a floating window.

B.3.5 Full Screen

[**10.7+ (Lion) Only**] This section only appears when Scrivener is running on Mac OS X 10.7 or greater, and impacts options pertaining to the OS X full screen mode, not Composition Mode (section B.5).

Always auto-hide toolbar in full screen mode You can opt to always hide the toolbar strip at the top of the project window when placing it on a full screen space. When hidden, the toolbar will be revealed along with the main menu by moving the mouse to the top of the display.

Hide binder and inspector when entering full screen mode In full screen mode, if previously visible, the binder and/or inspector will be hidden and replaced with slide-out version which appear as an overlay when they have been called upon by keyboard shortcuts, menus or buttons, or if the mouse is moved to the left or right hand side of the screen. Upon return to normal windowing mode, the original state of the sidebars will be restored if necessary. While in full screen, these sidebars can be accessed by moving the mouse to the left or right of the screen.

B.3.6 Fonts

Options for setting application interface fonts. If you wish to adjust the editor fonts, visit the Formatting tab (section B.8). These settings are optionally saved into preference Themes.

Header bar This font is used in the editor header bar to print the title of the selection you are currently viewing.

Binder Choose the font which will be used to display the titles of items in the binder and in Collections.

Outliner title Sets the display font for all text in the outliner except synopsis text.

> **Working with 2-byte Fonts**: When writing in a language that uses 2-byte fonts, such as many Eastern scripts, you should select a font here that natively supports the characters, rather than rely upon the OS X text engine to perform real-time substitutions. There is a limitation in the system which does not allow real-time substitution when font forcing is in use. The title will ultimately be displayed correctly when you view it later, but will not properly display the words you've keyed in until you reload the outliner session.

Synopsis Sets the display font for the synopsis text within an outliner.

Binder & outliner options Below these controls, you can also select formatting attributes to alter the appearance of container items in the binder and outliner, which can increase their visibility in long lists of items. Folders and document containers are treated separately. Note that if the fonts you have chosen above do not have a bold or italic variant, the conflicting option will be disabled. (In cases where you choose separate fonts for the binder and outliner, the formatting option will be enabled if the variant exists for either font, but the formatting will only apply to that element.)

These options will impact the outliner *only* if it is set to not show embedded synopses, or if the **Embolden all titles in outliner when showing synopses** option is disabled.

B.3.7 Customizable Colours

Custom interface colours are chosen using a column view. To alter the default colour for the desired interface element, first select the major category in the leftmost column and then the item itself from the middle column. Click on the colour well in the right column to assign a new colour, or click Use Default Color to reset the option to the application default. Since the program updates in real time as you alter the preferences, it is often beneficial to display the area you wish to customise, then adjust the colours while viewing the changes.

There are a few extra options located around the main table:

Use different colors in full screen [10.7+ (Lion) Only] When using the new Lion full screen feature, you can optionally choose to use a different set of colours for most, but not all, of the colour settings. When this checkbox is enabled, a "Full Screen" section will be added to the left-most column. Any changes made within this category will only be used when the project window is in full screen mode.

Use dark gray theme Adjusts a number of interface detailing to use a darker grey theme, which can improve the appearance of colour themes which are darker in mood. Additionally, if you use the Graphite system theme, you might prefer its graphite toned active header bar, which replaces the blue active header bar.

Use dark gray theme in full screen only [10.7+ (Lion) Only] When enabled, the dark gray theme will only be in effect when using Lion's full screen mode. In conjunction with using different colours to establish a darker overall palette in full screen, you could create an effective day mode vs. night mode.

Note: Some categories have identical settings, such as both the Editor and Composition Mode sections containing "Scrivening Titles". Where duplication exists, the reference tables (Table B.1) will list only the first entry.

Some background colour options also allow a texture. When you see the button **Choose Texture...**, you can set this particular element of the interface to use a tiling graphic for its background instead of a colour. Many tiling graphics can be found on the Internet for free. It is best to choose textures that are relatively small, as larger textures can slow down the interface or cause the computer to run out of RAM. To clear a texture you no longer want to use, just click the **Use Default Color** button. The elements which take a texture are:

- Editor: Text Background
- Editor: Page Background
- Editor: Fixed Width Background
- Composition Mode & QuickReference: Paper
- Composition Mode: Background

For a full list of all options, refer to the Appearance Colour Settings (Table B.1) table. These options are all saved into preference Themes.

B.4 Corkboard

Provides options and settings which impact the way the corkboard functions and appears.

B.4.1 Appearance

These options are all saved into preference Themes.

Draw lines in index cards Determines whether lines should be drawn in the text area of index cards (affects both the corkboard and the index card in the inspector).

Allow two lines in title areas When you require longer titles, the single-line truncation method that index cards use can be constrictive. Turning on this option will allow an additional line of word-wrapping before truncation occurs. Also see below for a font option to permit automatic shrinking of larger titles.

Display images as photographs Media files such as QuickTime movies and images are displayed by default as "Polaroids", and any document can be individually set to a

Table B.1: Appearance Colour Settings

Category	Interface Element	Description
Views	Binder Background	Background colour for the binder
	Search Results Background	Background colour for the search results collection and its tab
	Outliner Background	Background colour for the outliner
	Outliner Grid	The horizontal and/or vertical grid lines colour
	Media Background	Colour displayed around images, PDFs, movie clips, etc. in the editor
	Comments Pane	Background colour for the comments and footnotes pane in the inspector
	Scratchpad Text BG	Background colour for the scratch pad text-editing area
Editor	Text Background	Main editing background, or "paper colour"
	Page Background	"Gutter area" that is drawn behind the page when using **View ▸ Page View ▸ Show Page View**
	Scrivening Titles	Text colour for titles in Scrivenings mode as displayed in the editor when **View ▸ Editor ▸ ShowTitles in Scrivenings** is enabled
	Scrivening Titles BG	Background colour behind titles when enabled in Scrivenings mode; requires "Use title background color" to be enabled in the Formatting pane
	Fixed Width Background	Colour displayed around the pseudo-page in fixed width mode
	Notes Background	Document notes editing background, or "paper colour"
	Project Notes Background	As above, but for project notes; also affects the project notes window
	Footnote Comments BG	Background colour used for inspector footnotes
	Search Highlights	Highlight used to accentuate search results in the editor
	Invisible Characters	Text colour used to draw special symbols depicting invisible characters
	Insertion Point	Colour of the insertion point, or caret, that indicates where you are currently typing
	Text Selection	Background highlight colour used when selecting text in the editor; the **Use Default Color** button will pull the preference from your OS X system Appearance settings

Table B.2: Appearance Colour Settings Cont'd

Category	Interface Element	Description
	Current Line Highlight	Colour of the bar that highlights the entire line you are currently editing; visible when "Highlight current line" is enabled in Editor preferences
	Links	Colour of linked text
	Notes Text	Default text colour in the document and project notes
	Outliner Synopses	Text colour for the synopsis field in the outliner
Index Cards	Index Card Background	The "paper colour" for index cards. Will be blended with the label colour when **View ► Use Label Color In ► Index Cards** is enabled, so it is often best to choose a subtle shade
	Image Background	Background colour of the card when in graphical mode, used when no image exists or to fill in areas around an image that does not fill the card space
	Index Card Status Stamp	Base colour used to render the stamp text across the face of the card; the opacity can be adjusted in the Corkboard tab
Snapshots	Background	The "paper colour" for the snapshot text area. Note this will apply both to snapshots shown in the inspector and snapshots displayed in the main editor
	Deleted Text	Text colour used to indicate text that has been deleted in the difference between the snapshot and the current version
	New Text	Text colour used to indicate text that has been added in the difference between the snapshot and the current version
Revision Colors	*All elements*	Set individual colours for the revision level text tool
Target Progress Bars	Start Color	By default, progress bars blend through three colours; this is the colour of the progress bar when at its lowest
	End Color	The final colour of the progress bar
	Midway Color	The second colour of the blend. This will be the colour of the progress bar when it is at 50%.

graphical synopsis. Disabling this option will always show the standard synopsis index card on the corkboard.

Always show synopsis rather than image by default in inspector While the synopsis image will continue to be used in the corkboard with this option, the version of the card in the Inspector will always show the text synopsis instead. This can be useful if you are still using the text synopses as well as the image, as you can see both versions at once with a Corkboard and Inspector arrangement.

Arrange cards from right to left Will cause index cards be ordered from right to left, top to bottom. If you are accustomed to working in right-to-left languages, this will make the corkboard more intuitive.

Corkboard background Lets you customise the standard corkboard background. Select "Corkboard pattern", "Beige Graph Paper", or "Slate Graph Paper" to use one of the provided background patterns; "Custom color" to choose a background colour using the colour well to the right of the pop-up button; or "Custom background…" to select your own image file to use as the background. For better performance, use smaller, tiling textures rather than large images.

The **Freeform background** setting is identical.

Index card theme Lets you choose the line colours used in the index card. "Red and blue" divides the title from the text with a pinkish-red line and draws blue lines in the text area; "Blue and black" divides the title from the text with a blue line and draws black lines in the text area. These represent the most common colours of index cards in the real world. The "Rounded" theme displays cards in a more stylistic fashion, with rounded edges, lines drawn in grey, and labels represented as a colour block in the corner instead of as a pin. This setting will also impact the index card in the inspector (though it will always have square corners).

Position pins Choose whether pins on the corkboard (if displayed) should be drawn in the centre of the title area of each index card (which looks more authentic but can partially obscure the actual title) or in the top right corner. If using the rounded card theme, this option is not used.

Status stamp opacity Allows you to set the opacity of status stamps on the index cards within the corkboard. The stamp, unlike a real one, is drawn beneath the text, so even at full opacity it will not obscure synopsis text. This only impacts how boldly the stamp will be drawn.

Corkboard shadows Sets the intensity of the shadows around the cards and borders of the corkboard view. Slide all the way to the left to disable shadows.

B.4.2 Fonts

Options for altering the fonts used to draw index cards. To change a font, click in the preview panel to the right of the option you wish to adjust. This will present the standard font selection palette and any choices you make here will be applied to the preview panel immediately.

These settings are optionally saved into preference Themes.

Index cards title Lets you set the font used in the title area of index cards (affects index cards both in corkboard mode and in the inspector).

Photographs title If you have enabled the "Display images as photographs" option, the title of the image will be drawn below the thumbnail using this font choice.

Automatically shrink titles to fit with minimum font size As an alternative, or in addition to "Allow two lines in title areas", this is another option that can help accommodate longer titles in the index card, without excessive truncation. As the title grows beyond the size of the allotted space, the font size will shrink until it reaches the minimum amount specified. Once it reaches that point, it will truncate the title as per normal. Combined with two-line titles, very long titles can be used with little or no reduction in text.

Index cards text Lets you set the font used in the text area of index cards (affects index cards both in corkboard mode and in the inspector).

Small text Alternative font which will be used for *both* the title and text of index cards on the corkboard. The use of this font is determined by the project, not the preferences. To set this and other corkboard display options, see Corkboard Options (subsection 12.1.5).

Status stamp Lets you set the font family and typeface of the stamps that can be drawn diagonally over index cards in corkboard mode to show the status. Font size set here is ignored, as Scrivener automatically resizes text in stamps to fit the index cards.

B.4.3 Dragging and Double-Clicking

Options for how index cards react to click-and-drag events and to double-clicks.

Allow drop ons When enabled, lets you drop index cards on top of other cards to store them as children beneath that card. This action is similar to dragging items beneath other items in the binder.

Double-clicking on the corkboard background Adjust the behaviour of what should happen when double-clicking in a blank area of the corkboard.

"Opens the parent corkboard" will display the parent of the current corkboard. If the corkboard you are viewing is already at the top level of the binder, it has no parent and nothing will happen. The keyboard shortcut for this is ^⌘-R, and will remain in effect even when this option is not selected.

"Creates a new card" follows a common behaviour amongst diagramming applications, where you can indicate the position of a new card by double-clicking with the mouse.

"Does nothing" ignores all double-clicking on the corkboard background.

Always creates a new card in freeform mode When checked, freeform corkboards will ignore the above setting and always create a new index card under the mouse pointer when double-clicked. Uncheck this to have freeform corkboards follow the behaviour of the above setting.

B.5 Compose

These options affect the appearance and behaviour of the composition mode, which can be used to block out distractions and focus on writing.

B.5.1 Composition Mode Set-Up

Fade between modes Gracefully fades the screen between composition mode and regular editing mode.

Hide main window in composition mode Check this to hide the main window during composition. If this is unchecked, you can see the main window beneath the translucent areas of the composition background (unless you have set the opacity to opaque using the slider in the control strip), and you can also have the main window visible on another screen if you have a multi-screen set up. Checking this so that the window is hidden means that you could use the translucent areas of the background to look at the contents of the windows of other applications. Note that in some circumstances, having the main window visible in composition mode can slow down typing, so it is usually best to leave this checked as it is by default.

Use secondary screen if available If this is checked and you have more than one display, the composition window will be opened on the secondary screen.

Blank out other screens When using more than one display, select this option to reduce distractions from the other monitors. This merely draws the background colour

over the screen, and thus the monitor's energy saver will not kick in, so this is only useful if you'd rather not turn off your monitor.

Escape key closes composition screen When disabled, you will need to use either the View ► Exit Composition Mode command or its keyboard shortcut (⌥⌘–F) to toggle between modes.

B.5.2 Editing Options

Highlight current line Enabling this will place a background highlight colour beneath the current virtual line (not paragraph), making it easier to see where you are in your text. The current line highlight colour can be set below, in "Customizable Colors".

Show notifications when saving If you have Growl installed, this option will register a notice with the Growl bulletin system whenever you manually save the project, as there would otherwise be no indication of when save completed. Enabled only if Growl is installed and running.

Disable insertion point blinking When checked, the cursor will be always visible on the screen.

Use block insertion point of *n* pixels Adjusts the width of the insertion point from its default of 1 pixel. This will dramatically increase the caret's visibility; however, with variable-width fonts (like Cochin), this setting can produce odd results. It is best used with a fixed-width font like Courier.

B.5.3 Appearance

Paper margins Sets the margin spacing between the text and the "paper" edge. You can select a different distance for horizontal and vertical margins. The top and bottom margin will only affect the very top and bottom of the document on the page, and so will not be visible if you have scrolled into the middle of the file.

Scroller type Customise how the scroll bar looks in composition mode.

- *Regular scroller*: the standard operating system default scroller will be used. The appearance of this may be very similar to the "Minimalist scroller" option on OS X Lion, except in that its appearance behaviour will conform to the global Mac settings
- *Standard full screen scroller*: the scroller will look like the ones in the full screen modes of iPhoto and other iApps and will always be visible if there is enough text to warrant a scroller.

- *Auto-hiding scrollbar (uses text color)*: the scroller will take on the colour of the text if "Override natural text color" is selected within the "Customizable Colors: Text Color" setting (it will use black otherwise) and will only appear when the cursor is moved over it at the edge of the paper area.
- *Minimalist scroller (uses text color)*: uses a simple rounded grey bar, matching Mac OS X Lion's scrollbar appearance. This scrollbar, like the above, hides unless you move the mouse over the paper edge where it appears. Also as above, it will use any text colour override that is supplied.
- *No scroller*: composition mode will not feature a scroll bar under any circumstances if this is selected.

B.5.4 Customizable Colors

This table (Table B.3) only shows a sub-set of the available options, where they are unique or differ from the settings in the Appearance Colour Settings (Table B.1) table. Refer to it for any settings you are unsure of.

Table B.3: Composition Mode Colour Preferences

Interface Element	Description
Text Color	To activate this option, you will need to enable the "Override text color with color" checkbox. the text in composition mode will use the actual text colour—that is, the colour it is normally in the main editor. If this option is checked, you can choose a different colour for all text in composition using the colour palette. Note that this merely temporarily displays the text in the selected colour; the real colour of the text remains unchanged. This will also temporarily re-colour annotations, links, and linked note markers too, reducing their level of impact on your vision. The override colour can have an opacity assigned to it, which will allow you to blend the text colour with the page background.
Paper	The "paper colour" for the editing area, or pseudo-page, in composition mode
Background	Colour displayed around the pseudo-page in composition
Inspector HUD backgrounds	Synopsis, Notes, Project Notes, and Comments can all have their background colours set independently from the main preferences to match the composition colour scheme

B.5.5 Default View Settings

You can dynamically adjust most of the visual options while in composition mode, using the control strip at the bottom of the screen (you'll need to move your mouse down to the bottom if you cannot see it). Once you have your settings the way you would prefer for all projects (at least as a starting point), you can use the provided button here to save the following settings into your preference file as defaults:

- Paper position: There are three possible choices: "Left", "Center", and "Right". This is useful if you keep documents open in QuickReference panels or fade the composition background to view other applications alongside the pseudo-page.
- Default text zoom: Dynamically scale the size of the text font for composition mode. This will not impact the underlying font settings, letting you view your text larger or smaller only while in composition.
- Paper width: Adjusts the overall width of the virtual page. Move the slider all the way to the right to take up the entire screen or all the way to the left to use only a narrow column.
- Paper height: Brings the top and bottom of the page away from the edges of the display, keeping the text block centred on the screen. To set the paper height in the composition screen, with the Control Strip (section 17.2) visible, hold down the Option key and the "Paper width" slider will be replaced by "Paper height".
- Background fade: Lets you adjust the blend between the rest of your desktop and the chosen background colour. Move the slider all the way to the right to make the background opaque or all the way to the left to make it entirely transparent. Paper fade, the alternate form of this slider that appears when a backdrop is in use, is not saved into preferences.
- Typewriter scrolling: Enables this feature by default in new projects. Typewriter scrolling keeps the text you are currently editing in the middle of the window. It can be toggled on and off with **Format ► Options ► Typewriter Scrolling** (^⌘–T) while in composition.

B.6 Navigation

The navigation settings allow you to alter how Scrivener responds to the various actions you can take to navigate through your project, and how the program will react to your actions. There are numerous ways to move around in the interface—for example, clicking

directly on items, jumping straight to a specific place in your book via a menu, or using buttons to scroll through documents like a flip-book.

Unless otherwise indicated, these navigation options will apply to all methods of selecting an item in the interface.

B.6.1 Folders and Files

Treat all documents with subdocuments as folders When this is checked, any document that has subdocuments will act like a folder and will open in whatever view mode folders are set to use (the last group view mode selected). If this isn't checked, normal text documents or media files that have subdocuments (any type of document can act as a "container" for other documents in Scrivener) will be opened in single text mode.

Include folder text in Scrivenings (composite text mode) When clicking on a folder (or file group, if "Treat all documents with subdocuments as folders" is enabled), by default the text contents of that container will be included in the Scrivenings session at the very top. If you never use folder text, or document stack text, then disabling this will remove the empty entry at the top of the Scrivenings session.

Show folder text when selected from search results Overrides default behaviour of opening folders (and if "Treat all documents with subdocuments as folders" is set, file groups as well). When this is enabled, all container style search results will reveal their text when clicked on in the search results, rather than revealing their children in accordance with the selected view mode. This option is enabled by default, because generally speaking, most search results are based on the *text* of the matching item, not its descendent items, and so the reason the item is in the search list at all is because of its text.

Always create new items as siblings Ordinarily, when a container is selected, creating a new text item will place that item within it as a child, at the end of the list. (This will hold true for document groups as well as folders if "Treat all documents with subdocuments as folders" is enabled.) When this option is set, all newly created items will be created at the same level, below the selected item, no matter what the circumstances.

Folders are always created as siblings, even when this option is off.

B.6.2 Dragging

Option-dragging creates duplicates Enable this feature to cause the various organisational views to mimic the Finder in that when you hold down the Option key

while dragging items, it will create copies of those items in the location where you drop them, rather than moving the items.

In cases where the Option key is used to alternate or inhibit other behaviour, this will take precedence. For instance, if "Allow drop-ons" is turned on in Corkboard preferences, the ordinary use of Option to disable drop-ons will be inhibited, allowing you to create a duplicate of a card as a child of another card.

Collapse auto-expanded outline items after drag and drop When dragging items into the binder or outliner, if you pause over a collapsed container, Scrivener will helpfully reveal its contents so you can drop items several layers deeper than you could initially see. This option will close all automatically-opened containers after you have dropped the items. This setting impacts both the binder and the outliner view.

B.6.3 Lock in Place

Binder selection affects other editor when focussed editor is locked Typically, when a split is locked, if a binder click would have impacted that editor, the click action is blocked and nothing happens. Thus this does not include cases where the click would *not* have impacted the locked editor, such as when Binder Affects is engaged and targeting the unlocked editor, or the unlocked editor is not the active split. When this option is enabled, unless *both* editors are locked, a binder click will always load something, even if it has to redirect the click action to the inactive editor to do so.

B.6.4 Media

Rewind when paused by... Use the slider to set how many seconds Scrivener should rewind the media stream back, when using Auto-Rewind (subsection 14.3.3). This preference is not used unless the media player has been set to automatically rewind on pause.

B.6.5 Return Key

Ends editing synopsis in corkboard and outliner By default, the Return key will submit the edits being made to the synopsis in corkboard index cards and the synopsis outliner field. To add carriage returns to the synopsis, use ⌥Return. If this is unchecked, the Return key will add carriage returns, and you will need to use the Esc key to submit changes.

Creates new item in list, outline and corkboard views If this is checked, a new item is created whenever you hit the Return key in the outliner, binder, corkboard or

certain list views (such as the keywords list). Note that if you are editing the title of an item, hitting Return once ends the editing; hitting Return again will then create a new item.

B.6.6 Corkboard and Outliner Space Key Behavior

Space key opens selected document in... The spacebar can be used to quickly open selected items in Corkboard or Outliner. By default this opens the item in the editor, replacing the current view. If you would prefer it to match the binder behaviour, change this to "QuickReference Panel".

B.6.7 References and Scrivener Links

Create back-link references when creating Scrivener Links and references When you create links or references to internal items, Scrivener will automatically place a "back-link" to the originating document in the References table for the target. In other words, if you link *from* "Scene 81" to a character sheet named "Joseph", a reference link to "Scene 81" will be added to the "Joseph" document's reference table in the inspector. If you would prefer to maintain all of your references by hand, disable this option.

The following three settings have the same set of three options (with the exception of "Open newly created Scrivener Links in", which can be set to not do anything at all). You can choose to handle them by opening the target in a QuickReference Panel, which will preserve the current editing session, the Current Editor, which will replace the current editing session (this was the default behaviour in Scrivener 1.x), or the Other Editor, which will open the link in the other split, creating a new one if necessary.

Open references in... References to supported internal project items, which have been activated from the inspector reference table, will open using the instructions provided here. The default is "Other Editor".

In Full Screen mode, the Inspector panel will always open items as QuickReference panels. This is also true for any references activated from within a QuickReference panel itself, though References can be dragged from a QuickReference panel into the header bar of the main editor to load them manually, there.

See References (section 19.5) for full documentation on how to use this feature.

Newly created Scrivener Links refers to the creation of new documents with the ⌘–L shortcut, the wiki link style syntax, or the menu command **Edit ► Scrivener Link ► New**

Link.... New links have an additional option, "Do not open", which simply makes the link without opening the newly created document.

With the exception of "Do not open", creating a link *within* a QuickReference panel will ignore these settings and always open a linked item in a new QuickReference panel. Likewise, composition mode will use a QuickReference panel instead of the "Other Editor" (because it has no split view) and when opening linked items that are not text documents, since composition mode can only display text.

B.7 Editor

The options for the main text editor control how text is displayed (not to be confused with formatted) within the editor itself. For options pertaining to default text formatting, see the Formatting tab (section B.8); for options pertaining to text input and typography, see the Corrections preference tab (section B.9).

B.7.1 Editing Options

Smart copy/paste This functions in the same way as TextEdit. When enabled, extra whitespace around the words will be stripped out and the word buffered with a single space if necessary, cleaning the pasted text significantly.

Typing clears search highlights When using the Project Search feature, Scrivener will highlight the matching search results for you. If this is enabled, the highlights in the current editor will be temporarily removed as soon as you start typing. To restore them, select another document from the search result list and then use the editor's Back button (⌘-[) to return.

Highlight current line Enabling this will place a background highlight colour beneath the current virtual line (not paragraph), making it easier to see where you are in your text. To change the colour of this highlight, select Current Line Highlight in the Editor section of the Appearance tab (subsection B.3.7).

Use hyphenation Enabling hyphenation when using full justification can increase readability but at the expense of not being as "pure" to the actual text you have keyed in (the hyphens will not be literally placed into the text; they are only drawn on in real-time to improve word-spacing flow on a line). Consequently this is off by default.

This feature uses your system localisation preferences to determine optimum hyphen placement. If you are writing in another language and getting odd results, make sure your OS is set up accordingly.

Use fine kerning Uses a higher-quality text rendering model which reduces known screen artefacts, such as "text wobble" while typing, and ugly kerning at odd-number zoom settings. Disable this feature to marginally increase performance, if necessary.

Auto-detect dates, addresses, etc Switches on the operating system's data detection capabilities, which will attempt to scan for common text patterns and treat them appropriately. For instance if it finds a sequence of text that looks to be a postal address, it will give you the option to add it to your contacts list. Dates can be turned into events in iCal. This feature is only available to OS X 10.7 and higher.

Default Text Zoom Dynamically scale the size of the text font in the editor. This will not impact the underlying font settings, letting you use a larger or smaller font while editing. **With the exception of QuickReference panels, this only sets the default for future projects**. After project creation, each editor split can have its own settings and these settings will be preserved between editing sessions. QuickReference Panels take their default zoom level from this setting. Any panels that have already been opened in the current session will retain the old setting until the project is closed and re-opened.

Ruler Units Lets you choose the units (centimetres, inches, picas, or points) used by the ruler in the main editor, full screen, and QuickReference panels (available by hitting ⌘–R when the editor has the focus or by selecting Format ▶ Show Ruler). Rulers for the scratch pad and document and project notes use your system's settings at the time the project is opened.

Typewriter scroll line Selects the anchor point that Scrivener uses to reset the currently edited line's position within the editor. A default setting of "Middle of screen" thus positions the currently line in the vertical middle of the editor. See Typewriter Scrolling (subsection 15.3.4) for further details.

Disable insertion point blinking When enabled, the cursor will be always visible on the screen.

Use block insertion point of *n* pixels Set the width of the insertion point from its default of 1 pixel. This will dramatically increase its visibility; however, with variable-width fonts (like Cochin), this setting can produce odd results. It is best used with a fixed-width font like Courier.

B.7.2 Wrap to Editor Mode

There are two primary editor modes. Wrap to Editor works in a fashion familiar to most text editors. In this mode, text is wrapped according to the width of the editor view, or

to an artificial limit when "Use fixed width" is enabled. Wrap to Page mode emulates a printed page much like many word processors, though unlike word processors, it makes no attempt at accuracy, and is thus more an aesthetic choice for most.

Margins Sets the margin spacing between the text and the edge of the view. You can select a different distance for horizontal and vertical margins. The top and bottom margin will only affect the very top and bottom of the document in the editor; it will not be seen if you have scrolled to the middle of a long document.

Default Editor Width This value, in pixels, is used to determine the overall width of the project window when using the menu command Window ► Zoom. Scrivener will attempt to scale the window so that your preferred editor width fits in with the binder and, if it is open, the inspector. Click the Use Current button to capture the current width of the editor you were last using. Setting this to 0 (zero) will cause the editor to always take up as much space as possible (in effect, making Zoom act like Window ► Zoom to Fit Screen).

Use fixed width When this is enabled, Scrivener will also use the above setting to restrict the width of the editing "page" no matter how wide you make the editor itself. The area not used by this page will be filled in with a background colour, which you can set in the Editor section, under Fixed Width Background, of the Appearance tab (section B.3).

Full screen only [10.7+ (Lion) Only] Since full screen view often uses quite a lot more horizontal space, it can be desirable to enable fixed width editing, even if you do not prefer it in windowed mode. This checkbox will allow you that option.

B.7.3 Wrap to Page Mode

Show page view in new projects By default, new projects use Wrap to Editor mode. Enabling this will force Wrap to Page mode in all new projects.

Use facing pages in new projects Activates the two-page spread as the default for new projects. To change existing projects, use View ► Page View ► Two Pages Across.

Center pages Instead of always pegging the virtual page at the left side of the editor, no matter how wide, Scrivener will place the page in the middle of the editing view.

Disable Keep with Next At the expense of performance, you can deselect this option to preview manual orphan control. Keep with Next codes can be placed into a document using the Format ► Text ► Keep with Next command. Keep in mind that in many cases, paragraphs will actually be broken apart at different places in the final compiled result.

Show margin guides Will draw a border around the printable area within the virtual page.

Draw shadow around pages Adds a shadow around the virtual page, offsetting it from the background colour.

Spacing between pages When more than one page of text is displayed (or more than two, if using Two Pages Across), the number of pixels entered here will buffer each page from the one above it.

If you enter a value of 1 (one), and "Draw shadow" is enabled, a single line will separate pages horizontally.

If you enter a value of 0 (zero), and "Draw shadow" is enabled, pages will flow from one to the next (but will still have top/bottom margins separating them).

B.8 Formatting

B.8.1 Main Text Style

These settings determine how *new* documents will be configured. They will not adjust the formatting of documents you have already created, in any projects, so you needn't worry about these settings destroying your existing formatting.

Further, these settings can be adjusted *per project*. To change the settings for each project, select the menu command **Project ► Text Preferences...**. These will always be set to the application defaults shown here, when you make a new project. If you change the settings in the project formatting preferences, changes made to this tab **will not impact that project**.

It is best to visualise default formatting settings in the following order of precedence, starting with the "weakest" and ending with the "strongest":

1. *Application preferences* as set in this tab
2. *Project preferences* as set in **Project ► Text Formatting Preferences**
3. The document's text formatting itself will always override either of the above (and within that each paragraph can have its own ruler settings, and each character its own attributes).

To reset a document to either the project preferences (if applicable) or the application preferences, use the menu command, **Documents ► Convert ► Formatting to Default Text Style...**.

The top half of the window is dedicated to setting up the default settings for new text documents (files and folders). You will have access to the full standard ruler settings, as well as font controls via the fonts button (the italicised 'A'). All changes you make here will be immediately previewed against the provided sample quotation.

If you already have a paragraph set up the way you would like all future documents to appear, make sure that editor (and not the other split, if one is open) is active, and that the cursor is positioned within that paragraph, before clicking the **Use Current** button.

B.8.2 Formatting Editor Options

Writing direction In most cases, you will want to use the "Natural" setting, which is set up depending on your default system language. This should go from right to left, or left to right, automatically. If however, you do most of your writing in a language that is different from your system settings, you can manually override this here. Note you can also manually override individual documents as well in **Format ▶ Text ▶ Writing Direction**, this only sets the default. Authors working in both Arabic and German, for example, will probably want to leave this setting at Natural and change each document according to the language it contains.

Notes font The default font for all notes. This setting impacts Project Notes and Document Notes alike. Note that both of these are rich text fields and can be changed from this default. As with the main text editor, if you change this setting later on, old notes will remain as they were.

Underline links When disabled, Scrivener Links, hyperlinks, and notation links will not be underlined in the editor.

Do not color the text of inline notes (faster) Turns off the text colour for inline annotations—this option should speed up typing on slower machines. The bubble outline will turn into a filled background highlight, when this option is enabled.

Inspector comments font Sets the default font for inspector comments. Since these fields are rich text, this setting will only impact new comments. Inline annotations will always take on the styling of the text around them.

Use different footnotes font When enabled, the font within an inline footnote will be changed to this setting. It will still conform to the ruler settings of the text around it. To apply this setting to inspector footnotes as well, check the box labelled, "Use for inspector notes too", beneath it.

If you have change this setting halfway through a project, and would like to alter existing footnote formatting, the menu command, Documents/Convert/Formatting to Default Text Style will impact inline footnote formatting (subsection 15.4.5). Existing Inspector footnotes can be made up to date as well by resetting linked note formatting (subsection 18.3.3).

Terminate footnotes and comments before punctuation Some typesetting styles call for footnote markers to be placed before the terminating punctuation for a sentence, rather than after it, or for when individual words are referenced rather than the entire sentence. Select this option if you require this format. You can always manually define where the marker will be placed by selecting the range of text you wish to comment on, prior to making the note. This only alters the behaviour used when selection is made automatically in lieu of a selection.

B.8.3 Scrivenings

The first option, *Separate with single line breaks* sets a lower-profile and more vertically accurate divider marker, rather than the default full-width divider. This option is useful when page formatting must be precise, such as in scriptwriting, or if you usage of scrivenings tends more toward very small chunks rather than section-oriented chunks. Some may also find it to be more visually appealing than the divider line.

> **Working with 2-byte Fonts**: When writing in a language that uses 2-byte fonts, such as many Eastern scripts, you should select a font here that natively supports the characters, rather than rely upon the OS X text engine to perform real-time substitutions. There is a limitation in the system which does not allow real-time substitution when font forcing is in use. The title will ultimately be displayed correctly when you view it later, but will not properly display the words you've keyed in until you reload the Scrivenings session.

When Format ▸ Options ▸ Show Titles in Scrivenings is enabled, the following font settings and background colour option will adjust how the text used to show titles in the edit scrivenings session will appear:

- *Center*: the title will be centred in the editor, rather than left-aligned
- *Underline*: standard single-stroke underlining will be used on the title
- *Use title background color*: when checked, the title background colour will be used to highlight the title. This can be set in the Appearance pane (subsection B.3.7) for the standard editor, and the Full Screen tab (subsection B.5.4) for composition mode.

B.9 Corrections

The Corrections tab contains optional typing aids, such as typographic enhancements (superscripted ordinals, em dashes, smart quotes, etc), word completion, and control of the spelling engine. Not all of the places you can type in Scrivener will support these aids. The main text editors (standard, composition mode, QuickReference) support every option. For more information on corrections and auto-completion, read Auto-Completion (section 15.7).

B.9.1 Spelling

Read more about Spell Checking (subsection 15.2.3).

Check spelling as you type in new projects Displays red underlining for words that are not found in the built-in dictionary, as you type. This setting only impacts future projects. To change the behaviour of existing projects, toggle this feature on in the Edit ► Spelling and Grammar menu.

Check grammar with spelling in new projects Displays green underlining for words and phrases which the built-in grammar checker deems to be in poor style or in violation of grammar rules. Since this is a subsidiary function of the spell check engine, spelling must be checked (above) for this to be active.

Correct spelling errors as you type Will correct common spelling errors and typos automatically. This feature can be trained with new words you've added to the dictionary.

B.9.2 Auto-Capitalization

Fix capitalization of sentences Will fix letter case issues if a lowercase letter follows a period and a space. You may need to temporarily switch this off to correctly key in some phrases, such as "10 a.m. to 12 p.m.". In that case the period after the "a.m." would cause the word "to" to be capitalised incorrectly. Disable the option to key in the phrase, and then enable it again.

Capitalize 'i' Will automatically capitalise the letter 'i', if typed by itself.

B.9.3 Auto-Completion

Word completion is usually used to speed up the entry of places and names and other common project specific terminology. However, it can be set up to use the entire language dictionary and attempt to complete every word you start typing. By default this is not enabled as most users find it too intrusive.

Suggest completions as you type Enables auto-completion in general. When this is unchecked, no auto-completion will ever be dynamically performed, but you can always manually request word-completion by pressing the Esc key, or ⌘–.. Words will be presented to you in a list of narrowing specificity the more you type. Once you've narrowed it down to the right word, or select one from the short list, you can tap Enter to select it and proceed on with the next word. This also pulls from your project's auto-completion list, where you can add frequently used phrases in your work to speed up text entry.

In script mode only Disable the auto-completion feature unless the editor has been set to script mode.

Only suggest completions from custom auto-complete lists Will restrict the auto-completion list to only those words that you have specified in Project ▶ Edit Auto-Complete List.... When disabled, will attempt to find words using the exhaustive built-in language dictionary.

B.9.4 Substitutions

Enables a wide array of commonly used symbols and typographic conventions, by detecting when their use is appropriate. Not all text entry areas are capable of using substitutions, but they will always be available in the main text editors.

If you are using the MultiMarkdown export system, it is best to leave typographic substitutions disabled, as it will handle all of that automatically after export.

Most of the options are self-explanatory. Those requiring explanation follow:

Use smart quotes Will convert inch and foot characters into typographic quotes (also referred to as "curly" quotes) as you type, according to your system language settings. To customise these, click the System Text Preferences... button that is provided at the bottom of the window.

Replace double hyphens with em-dashes This uses the Mac option to toggle this behaviour, which also includes ellipses on 10.8 systems and above. Subsequently, the following option, **Replace Triple periods with ellipses**, is only of use to those using older versions of Mac OS X.

Disable smart quote, em-dash and ellipsis substitution in script mode The three options above this one will be ignored in script mode.

Superscript ordinals When numbers are typed with a following ordinal, the ordinal will be superscripted and set to a smaller font.

Automatically detect web addresses If you start typing in a web address, Scrivener will automatically generate a web link for it so you can click on it and open the link in your web browser.

Automatically detect ((Scrivener links)) Scrivener document titles can be linked to by surrounding the title in double-brackets. For more information on this feature, see Linking Documents Together (section 9.5).

Symbol and text substitution Utilises the system level substitution engine for OS X 10.6 and higher. To modify the list, click the **System Text Preferences...** button.

B.10 Import and Export

Scrivener supports a wide variety of file formats, both for import into a project and for exporting or compiling. This pane has been separated into three tabs, which will be covered in the following sections.

At the bottom of the pane is a section for using higher quality, but slower, import and export engine for .doc, .docx, and .odt files.

Enable enhanced converters for Microsoft Word and OpenOffice documents When enabled, the Aspose engine will be used for all import and export activity, including drag & drop into the binder, normal file import and export, and compilation. This will only impact the processing of .doc, .docx and .odt type files. The quality of the import and export should be markedly improved, though at the expense of speed. If you are using an older computer and find the speeds too slow, it might be easier for you to convert the imported documents to RTF first, or to export and compile as RTF.

B.10.1 Import Options

Only import files fully supported for opening in Scrivener's editor By default, Scrivener will allow you to drag file types into the binder[3] even if it cannot internally display them. Scrivener will store unsupported files with the rest of the project, but will not be able to display them, though with Mac OS X 10.7+, you'll see a Quick Look preview of the file, if it provides one. You will still be able to load and edit them using external applications.

This option prohibits importing unsupported file types. If any have been imported into projects in the past, they will remain.

[3] You can only ever drag text files, or folders with text files, into the Draft

Rich Text Files (RTF, Microsoft Word, and OpenOffice) These two options set how RTF comments and footnotes will be converted to Scrivener's notation format. By default, they will be formatted as linked notation. If you prefer inline notation, check off these options. Since Scrivener exports RTF comments and footnotes when using the external folder sync tool, this will also govern how that round-trip occurs.

Owing to the fact that RTF does not have a separate way of addressing these, there is no way for Scrivener to retain a mix of types. When importing or syncing, they will all be converted one way or the other.

Plain Text When importing plain-text files (usually they have a .txt file extension, but Scrivener will assume any file without an extension is a plain-text file) you can have Scrivener either apply the current default styling preferences, or optionally use a special font for these files.

Convert brackets to inline notes when syncing plain text files This setting will instruct all of the synchronisation methods found in the File ▶ Sync ▶ sub-folder, which use plain-text (either inherently or by preference) to export your inline notation enclosed within special brackets. These brackets will be searched for when you sync back, converting them into notation format again in Scrivener. The brackets used are:

- ((Text of inline annotation))
- {{Text of inline footnote}}

You can type in these brackets in the plain-text files to add new notation ranges to your text while on the go. Because this only works with inline notation, you will need to convert your inspector comments and footnotes to inline if you intend to take advantage of this feature.

HTML You can set how HTML files will be handled if you drag them into Scrivener. The WebArchive (Safari format) option will package the document into Apple's WebArchive format, which is viewable in many Macintosh applications. If you plan to eventually make project resources available to any platform, you should use the Text option instead. Text will preserve basic formatting and allow editing, but complex page formatting will be lost.

Optionally, you can also choose to convert imported WebArchives downloaded web pages to text as well.

OPML When OPML files are dropped into the binder, Scrivener will attempt to convert these outlines into Scrivener outlines. This can be useful if you do your initial brainstorming in a dedicated outliner application. Some of these applications support attached notes to each outliner header. If notes exist, Scrivener can be instructed to apply these notes either to the Synopsis alone, the main text, or the document notes for that heading in the outline. Note that in the case of the last two options, the synopsis will still be set automatically to the first few hundred characters of the note.

Enable **Create folder to hold imported OPML items** if you want dropped OPML structures to be contained inside of a new folder, rather than directly into your existing Binder structure.

Import first lines of Scapple notes as titles only When dragging Scapple notes into Scrivener you can choose to have the first line of the note stripped out of the main text and synopsis, reserving it strictly for the binder title.

B.10.2 Plain Text Import Types

Scrivener supports a number of plain-text extensions out of the box:

- .txt
- .xml
- .log
- .do
- .tex
- .md
- .mmd
- .markdown

It also will treat any file as plain-text if it is lacking an extension. You may have some files which you would like to be treated as plain-text in addition to the above list. You can use this table to add your own extensions to the recognised file list. There is no need to specify the 'dot' in the extension list. This will not *convert* a non-text format to plain-text. For example, if you add 'xhtml' to the list, it will import the raw XHTML code into the text editor (incidentally overriding the standard HTML importer which *would* attempt to convert it). So take care to not include extensions which might be used for binary or codified data, unless that is what you really want to do.

B.10.3 Export Options

Rich Text Files (RTF, Microsoft Word, OpenOffice)

Resolution for PDF images converted to PNG Since these formats do not support embedded PDF files, any PDF graphics that have been embedded in Scrivener's text editor will be converted to PNG raster graphics upon compile. This option lets you determine what resolution the PNG should be set to. The default of 300 dpi (dots per inch) is a common standard in the publishing industry and will ensure a large enough intermediate format for converting vector graphics to raster graphics with a balance between quality drop and file size. You can tune this upward if the resulting graphics are still not to your requirements and appear blurry, or if you have been instructed otherwise by the publisher.

Microsoft .doc This setting now only impacts export when the new improved converters cannot be used. In most cases, you will want to leave this set to the default, RTF-based .doc file. Scrivener's enhanced RTF export engine supports a great many features that OS X's built-in exporter does not. However, if you are working with an application that cannot read RTF files at all, and cannot use the enhanced converters, you will want to change this option to use the standard OS X exporter.

Use Word-2011 compatible copy If checked, copying and pasting from Scrivener into Word 2011 will use true footnotes and comments. This is needed owing to changes in Word 2011 (which now prefers RTFD pastes even if an RTF alternative is supplied), but is off by default because RTFD is useful for pasting into other Apple and Cocoa based programs.

Include image file names for Nisus Writer (RTF only) If your primary word processor is Nisus Writer (Pro), this option will add the names of images to their internal meta-data. This option has no impact in other word processors to date, so it is safe to leave it on unless you encounter difficulties with it.

HTML

These settings will affect the manner in which compile produces .html and .webarchive files, as well, but not the MultiMarkdown HTML format, as that uses its own system.

Document type "Document type" refers to the web standard, or DocType, that should be used during export. If you are unfamiliar with web standards, chances are the default setting of the older "HTML 4.01 Strict" will suffice. If you intend to publish your writings within a modern blogging system, however, in most cases using one of the modern XHTML formats will be more compatible.

Styling Styling can be set to three levels. Embedded styling will declare CSS styles at the top of the exported file, allowing you to easily adjust them with your own custom stylesheets. Inline CSS puts the styling directly into the HTML elements themselves, and so is useful in situations where you wish to override default stylesheets. Finally, "No CSS" will export devoid of any styling. In many blogging and content management systems, all presentation formatting is already handled for you, and this will be the best option.

Preserve white space Attempts to convert your ruler settings into the appropriate CSS style codes.

MultiMarkdown export language The MultiMarkdown engine will handle the conversion of typography for you, which is particularly useful if you intend to keep your source documents ASCII compatible. These are language specific, so choose the language converter most appropriate to what you are writing in.

JPEG images compression Sets the amount of compression used when saving JPEG images embedded in text files, for export and compile. The minimum (left) slider position will result in the best quality images, at the expense of file size (better for print). The maximum (right) slider position compresses images as much as possible, resulting in smaller file sizes (better for online or e-publishing).

B.11 Backup

Scrivener has a fully automated backup system. The options in this tab will control the various aspects of it.

Turn on automatic backups Enables the automated backup system. It is best to leave this turned on, unless for performance or security reasons you wish to control the process manually.

Back up on project open Whenever a project is opened, a backup will be created before you can begin working. This can slow down load times with large projects.

Back up on project close Whenever a project is closed (either directly or via application shutdown), a backup will be created. This can slow down close times in large projects.

Back up on each manual save Will create a backup copy whenever you manually save a project with ⌘-S. Note this is not the same as Scrivener's auto-save feature, which by default is triggered after every two seconds of idling. Backups will only be triggered if you select the menu command manually.

Compress automatic backups as zip files (slower) If you are storing your backups on a network drive, or Internet synchronised service like DropBox, this option will not only save space but will protect the project format from the sorts of damages which can occur when lots of files are transferred over the Internet. However, it will adversely affect the performance of your backups, as Scrivener not only has to collect and assemble a copy of the project, but compress it as well, before letting you return to your work. With very large projects, this can take many minutes to complete, and turning it off can save a lot of time.

Use date in backup file names In addition to a numbering token, which Scrivener uses to keep track of the rotation, the date and time of the backup will be included in the file name, making it easier to sort by recency in the Finder. It also means each backup filename will be unique, which could be of use with an external backup system.

Only keep *n* most recent backups To avoid the proliferation of hundreds of backups, a limiter is employed. Use this option to have Scrivener restrict the number of backups to a defined amount. Once this amount is reached, it will rotate through the copies, overwriting the oldest available. You may find this needs adjusting if your work habits cause backups to overwrite each other too quickly. It's generally a good idea to have three or four days of backups available.

Disabling this option will cause Scrivener to never remove old backups automatically.

Backup location The default location stores your backups in your user Library folder, under Application Support/Scrivener. If you would prefer these backups be kept somewhere more visible, or on a synchronised, external drive, or secured area, click the **Choose...** button and use the folder navigator to select the destination folder. If you choose a location that later becomes invalid (a common example is external drives), Scrivener will immediately warn you as soon as it tries to back anything up.

As a convenience, you can quickly access the backup folder by clicking the **Open backup folder...** button.

Appendix C

Placeholder Tags

For a complete list of all placeholder tags available in Scrivener, see **Help ► Placeholder Tags List...**.

Appendix D

Tips and Tricks and Trouble

D.1 Getting Support

As a registered user of Scrivener, you are eligible to receive technical support for any enquiries you may have, from problems to general questions. You are also invited to participate in our discussion forums, where a vibrant community of other Scrivener users can often offer advice and help. For all avenues of support, please visit our main support web page[1] for details on where to proceed and which e-mail address to use.

D.2 Using Scrivener in a Secure Environment

> **Disclaimer**: This advice is designed to make you aware of all known Scrivener data leaks, and does not constitute security advice. Consult with your security team on the viability of the listed strategies and adjust them as necessary to produce results which conform to whatever guidelines are required of you. Literature & Latte make no claims about the security of Scrivener in confidential environments.

If you work in a field that requires confidentiality and/or encryption, it may be possible to convert Scrivener into a semi-secure application. Be aware that Scrivener is not coded with maximum security in mind, and unless you are using third-party tools and hardware to protect your active RAM and hardened equipment, security is only going to apply to the data which is stored on the disk. Your project will be held (in full or in partial) in unencrypted RAM whenever you are working on it.

If you are not using FileVault or some other technology to protect your user folder, you can create pockets of encrypted storage in your user folder with Apple's Disk Utility application, which comes installed on every Mac and can be found in the Utilities sub-folder in Applications. Use this tool to create encrypted disk images to the size and protection level that you require, and then store your Scrivener projects on these disks. Documenting the production of these disk images are out of scope for this manual. You will find full documentation in Apple's help files for Disk Utility.

[1] http://www.literatureandlatte.com/support.php

Further steps will need to be taken to ensure that data in these projects are not accidentally disseminated. By default, Scrivener creates a backup of your project, every time it is closed, to your Library folder. You will want to adjust this location to reside within an encrypted drive as well (it need not be the same one). Do note that sequential backups can take up a lot of space over time and prepare an encrypted disk large enough to accommodate at least five copies of each project you are working on with plenty of room to spare. You can alter the location of automated backups using the Backups preference tab. The default location is:

```
~/Library/Application Support/Scrivener/Backups
```

If you are working in a mixed environment, or only wish to protect a single file, you might instead opt to exclude the sensitive project from the backup system individually. This can be done with a menu toggle, **File ► Back Up ► Exclude from Automatic Backups**. You will now need to maintain your own manually scheduled backups.

Finally, the Scratch Pad also stores information in a non-encrypted area of your Documents folder. Either take care to refrain from putting sensitive data into the Scratch Pad, or use the General preference tab to relocate its storage location to an encrypted disk image. The default location for Scratch Pad files is:

```
~/Documents/Scrivener Scratch Pad Notes
```

When Scrivener panics because it cannot write to the disk, it will terminate your editing session and dump anything in RAM to your Documents folder. These files will be placed in a folder named for the project itself, using the format, "Project Name.scriv Recovered". If you lose the connexion to the encrypted disk or for some other reason find yourself dumped out of your project, you will be warned with a dialogue box, and you should take steps to move the files from Documents to the encrypted area and digitally shred the originals.

If you are using FileVault or a disk-level encryption tool, then your entire user folder will be encrypted, and you can use Scrivener in its normal configuration. It does not save any project material to places outside of your home folders, including diagnostic and error recovery data.

D.3 Hidden Preferences

Hidden preferences are features which have been developed for niche purposes or to modify the behaviour and appearance of the application in ways which are not supported. If you use these, and encounter errors, please do not report them as bugs. To set hidden

preferences, you will need to use Terminal.app to copy and paste in command line codes. If you are not familiar with using Terminal, please reference how to use it in its documentation, and be aware that UNIX level access to your system inherently has less safe-guards than the standard OS X interface (for instance, files can be deleted *en masse* and without any confirmation whatsoever, in a way which completely bypasses the Trash). Always double-check your commands and make sure there are no typos.

The basic syntax for adding preferences to OS X applications is as follows:

```
defaults write application.address PreferenceCode Value
```

For each hidden preference, you will be supplied with the complete command required to enable it. To remove the preference, you will need to use a slightly different command:

```
defaults delete application.address PreferenceCode
```

No value is required when deleting a preference setting as the whole key will be removed from the preference file. For example, to reset Scrivener's built-in divider with to 1px borders instead of thick Jaguar era split dividers, type in:

```
defaults delete com.literatureandlatte.scrivener2.plist \
SCRNoSinglePixelSplits
```

In this last example, it has been split into two lines with a backslash on the end. The backslash instructs the command line that there is more to follow on the next line, so it is important to leave it as-is. It is always a good idea to close Scrivener while executing these commands, or at least restarting it after they have been executed.

Difficulties Editing Preferences on 10.9+: With the release of 10.9, Apple changed the infrastructure for how preferences are processed behind the scenes. The result of this is that preferences are now cached separate from, and used in preference of, the disk copy. What this means in practice is that you may find using these console commands has no effect, and that if you examine the .plist file in a text editor, or use the "defaults read" command, that the setting you just added does not exist in the file, even if you double-checked for its existence prior to launching Scrivener. This is because the cached copy has replaced the disk copy. If you experience it, you can often cause the cache to be updated by reading in the preferences after writing them. To do this, after the execution of any of the listed commands, issue the following in the command-line:

```
defaults read com.literatureandlatte.scrivener2.plist
```

> Your Scrivener preferences will be dumped to the terminal window as feedback when it works properly. If upon restarting Scrivener, the manually set preference still does not stick, you may need to log out and back in, and then try again with the "defaults write . . . " command, followed by the read command, above, before starting Scrivener for the first time in that session.

D.3.1 Hide Binder Icons

To disable icon drawing in the binder and produce a text-only outline style, paste the following into Terminal:

```
defaults write com.literatureandlatte.scrivener2.plist \
SCRHideBinderIcons 1
```

This will impact the entire application. However if you would like to disable icons for only some projects, use the following format:

```
defaults write com.literatureandlatte.scrivener2.plist \
[MyProjectFileName].hideBinderIcons
```

The project file name should not include the extension. So for instance, to hide binder icons for MyNovel.scriv project:

```
defaults write com.literatureandlatte.scrivener2.plist \
MyNovel.hideBinderIcons 1
```

Finally, you can disable only the text file and text file stack icons by adding the following command:

```
defaults write com.literatureandlatte.scrivener2.plist \
SCRHideOnlyTextIcons 1
```

For this to work, you will need to have SCRHideBinderIcons set, otherwise it will do nothing.

D.3.2 Adjust Table of Contents Indent

The ability to copy binder items and paste them as a rough Table of Contents can be adjusted by setting the default indent, in points. By default, items relative to the top level of the ToC will be indented half an inch (36 points) for each level. To change this, copy and paste the following into Terminal:

```
defaults write com.literatureandlatte.scrivener2.plist \
SCRToCIndentLevel 18
```

Where the final value is the number of points you wish to indent by for each level. This setting will also impact the offset of the secondary tab, which can be used in the link prefix to offset numbering from the title.

D.3.3 Use Thick Dividers

Since Leopard, the use of single-pixel dividers between interface elements has become the visual standard. Some prefer the older look where the divider is an element that can be grabbed along any point easily, rather than hunting for the pixel, or using the drag strip at the bottom. To set Scrivener to use thicker dividers, paste this command into the Terminal:

```
defaults write com.literatureandlatte.scrivener2.plist \
SCRNoSinglePixelSplits 1
```

D.3.4 Allow comments and footnotes in tables

Currently comments and footnotes are not supported in tables because clashes during the export process can cause errors to get thrown. A new hidden default has been added, however, so that users who need footnotes or comments in tables can enable this feature at their own risk—if you do so and encounter errors, please report the errors and the steps to reproduce to us, so that we can find a better solution in the future and enable footnotes in tables by default. To turn on the hidden preference, in Terminal, paste:

```
defaults write com.literatureandlatte.scrivener2.plist \
allowCommentsInTables 1
```

D.3.5 Invisible Links in PDFs and Printouts

When generating a PDF or printing via Compile, hyperlinks (this does not apply to Scrivener links) will always be printed in standard underscored blue. If you would prefer them to be invisible, you can set the following preference:

```
defaults write com.literatureandlatte.scrivener2.plist \
SCRDoNotFormatHyperlinksForPrinting 1
```

D.3.6 No Automatic Binder Selection

If you prefer the way Scrivener 1.x worked, in that the Binder never selected and revealed files for you under any circumstances, you may use this preference:

```
defaults write com.literatureandlatte.scrivener2.plist.plist \
SCRNeverAutomaticallyRevealInBinder 1
```

When this is active, Scrivener will never adjust the Binder selection without you doing so specifically with the mouse, arrow keys, or **Reveal in Binder** command. Most prominently, it will disable automatic selection and reveal (outline expansion in order to show a nested item) under these circumstances:

- Switching between collections
- Closing collections
- Next/previous document
- Moving out of full screen.

D.3.7 Disable Hyperlinking to Internal References

When creating Scrivener Links to imported web pages, the original URL will be hyperlinked to when compiling to formats which support hyperlinking. This behaviour can be disabled with:

```
defaults write com.literatureandlatte.scrivener2.plist \
SCRDoNotConvertWebDocLinksToHyperlinks 1
```

When this is active, these Scrivener Links will simply be omitted. This settings will only toggle this behaviour off for internal links to imported web pages. If you wish to disable compiling for all Scrivener Links in HTML output, do so in the HTML Options compile pane (section 24.16), and the Transformations compile pane (section 24.15) for all hyperlinks in any format that supports linking.

D.3.8 Filter Front Matter Items when Compiling

When compiling with the Filter option engaged, the contents of the Front Matter folder (if one is selected) will always supersede the filter and be ignored by it. In this way, you can use the filter feature to selectively compile your draft contents whilst still presenting the compiled document with the title page, table of contents or whatever else without making sure these elements also match the filter criteria.

If you would prefer the filter to remove front matter items that do not match its criteria, then enable this option:

```
defaults write com.literatureandlatte.scrivener2.plist \
SCRAllowFrontMatterFiltering 1
```

D.3.9 Do not auto-hide find panel when searching

Ordinarily, when pressing **Enter** to search from the standard **Edit ▶ Find ▶ Find...** tool, the find box will disappear and you can continue to search for matches using the **⌘–G** and **⇧⌘–G** shortcuts. If however you prefer to use the mouse you can disable this behaviour with:

```
defaults write com.literatureandlatte.scrivener2.plist \
SCRSuppressFindPanelAutoClose 1
```

D.3.10 Disable underscores in bookmark labels

Note to Nisus Writer users: There is currently a bug in Nisus Writer whereby internal links do not work when pointing to bookmarks with underscores in them. By default, however, Scrivener replaces any whitespace in file names with underscores when generating bookmark names. You can turn this behaviour off and ensure that bookmark names contain no underscores using the hidden preference:

```
defaults write com.literatureandlatte.scrivener2.plist \
RemoveUnderscoresInRTFBookmarks YES
```

Nisus users wishing to use internal links are therefore recommended to turn this hidden preference on until the bug is fixed in Nisus. Users of other word processors should never need to touch this hidden preference.

D.3.11 Always use Notification Center

On Mountain Lion, if Growl isn't installed, target notifications will use Notification Center. If you have Growl installed, you can still force Scrivener to use Notification Center by using the hidden preference:

```
defaults write com.literatureandlatte.scrivener2.plist.plist \
AlwaysUseNotificationCenter 1
```

D.3.12 Default Binder Width

In OS X 10.7+ Full Screen mode, the Binder and Inspector can be set to slide when the mouse is moved to the edge of the screen, rather than taking up window space. If you would like to change the default size of the binder width, you can use this setting:

```
defaults write com.literatureandlatte.scrivener2.plist.plist \
BinderSlideInWidth PIXELS
```

Where "PIXELS" is an integer specifying how many pixels wide the binder should be.

D.4 Troubleshooting

D.4.1 FAQ

For an up-to-date "Frequently Asked Questions" list, visit our knowledge base[2].

D.4.2 Some compile pages start too far down on the page

If some of the pages in your printed or compiled copy start further down on the page than you anticipated, and you can find no extra carriage returns in your draft, you might have the page padding feature enabled in the Formatting compile pane. For each type and level, check on the right-hand side of the formatting area for a Page Padding value of greater than zero. This will insert padding on pages that follow a page break in the compilation.

D.4.3 Project Search doesn't find text it otherwise should

Scrivener keeps a search index of everything you import or create in your project. This makes searching speedy and efficient, but it also means that in some rare cases, the search index can become "desynchronised" from the actual material in your project. This usually happens when the project file has been manipulated from outside of Scrivener. If you feel the search index does not accurately reflect the content, or your otherwise suspect that the project might have been damaged by an external application, you can manually rebuild the search index by holding down the **Option** key and selecting **File ▶ Save and Rebuild Search Indexes**.

D.4.4 Smart quotes and quote conversions don't work

Scrivener uses the Snow Leopard system language preferences. If you are using Snow Leopard, and have your system Text preferences set up to use straight quotes for smart quotes, then it will use straight quotes, and will not "see" other quote types when using the **Format ▶ Convert ▶ Quotes to Straight Quotes** command.

[2] http://help.literatureandlatte.com/help/kb

D.4.5 Scrivener icon turns into a question-mark in the Dock

This usually happens when Scrivener has either not been properly installed on the system, or the application was dragged from the downloadable DMG or CD product, instead of the Applications folder. The Dock does not store your application, but merely a reference to it. If the original target existed on a drive that is no longer attached to the computer, it will not know where to look, and produces a question-mark icon. To fix it, drag the question-mark icon out of the Dock, and in Finder press ⌘-⇧**A** to open your Applications folder. Locate Scrivener, and drag it from there into the Dock.

If you cannot find Scrivener in the Applications folder (or wherever else you thought it had been installed), then it likely never get properly installed in the first place. Follow the instructions in Installation and Upgrading (chapter 4) to fix the problem.

D.4.6 Save PDF to Scrivener stops working

In some cases, the link between the PDF service and the Scrivener application can get broken. Most often this happens when the application has been updated or re-installed. If you've tried using the "Save PDF to Scrivener" choice from the print dialogue of any application, and it doesn't work, then open the "PDF Services" folder, in your home Library folder, and delete the alias called "Save PDF to Scrivener". Restart Scrivener, let it create a new copy, and try again.

D.4.7 Starting Up without Prior Projects

If a project has crashed, and causes Scrivener to crash again each time you launch it because it keeps attempting to open this project, you can temporarily force Scrivener to ignore any projects scheduled to be opened, by holding down the Shift key while the program starts up.

This trick only works in Snow Leopard and above.

Appendix E

Project Bundle Format

A Scrivener project is stored on your computer as a cohesive and integrated set of standard folders and files. This internal structure is designed to be logical and accessible, in the rare case that you need to salvage material out of your project when Scrivener is not otherwise available. On a Macintosh, the operating system will obscure the fact that this is a folder from you, and display it as though it were a single file. Since this is not a feature of the Windows operating system, the contents of the project will always be visible to you. If you are using a Mac, you'll need to right-click on the project file and select "Show Package Contents". The remainder of this section will discuss the contents of the .scriv package as though it were a normal folder.

The entire folder structure is considered to be a part of the project, and should never be separated from the rest. Consequently, it is often best to use an archive method to back up your project, such as Zip compression. This increases the portability of the project in convenience and size, and protects it from accidental damage. This should be done by zipping the container .scriv folder (or package).

On the flip side, you shouldn't use the .scriv folder as a place to store material. While this theoretically won't hurt anything, we can't guarantee that it will remain safe there, as this entire folder and its contents are curated by the software. If you wish to organise external material along with your project, we recommend creating a folder to contain both the .scriv folder and whatever other documents you wish to store. This warning is naturally more pertinent to Windows users.

At the top-most level, the folder will be named according to the project name you chose, and have a ".scriv" extension added to it. Beneath that level, the project.scrivx file is the master file which defines the structure of your project[1]. Every single entry in the binder, your keywords, your print settings, all of these and more are saved into this XML file.

A new project will have anywhere from two to five subfolders at this top level, depending on which features have been used in the project.

[1] In older projects on Windows, the name of the .scrivx file may match the folder name.

- Files: where most of your actual data will be stored. Here you will find all the pieces of your draft, your research material, imported media, and so forth.
- Icons: if custom icons have been saved into the project, they will be stored here. This folder will be created as necessary.
- QuickLook: on a Mac, this provides information to the Finder's Quick Look preview function. It gives you a quick overview of the project's contents in an indented outline format. Windows does not use this folder, and unless you've opened the project on a Mac, you won't see it.
- Settings: project level settings and appearance settings files will be saved into this folder, as well as computer specific information used to lock the project while you are editing it.
- Snapshots: this optional folder will be used to store snapshots, and may not exist until snapshots have been created within the project.

E.1 The Files Folder

This folder contains the following items:

- binder.autosave: a redundant backup of your primary .scrivx file, at the top level. If the primary file becomes corrupted or goes missing, this can be used in an attempt to recover the project. A new copy is created every time Scrivener auto-saves.
- binder.backup: a third level of backup protection, this file is only created after you successfully open a project, ensuring that it is a valid copy of the primary .scrivx file.
- Docs: this folder stores all of your binder data, from draft sections to imported material. If it is listed in the binder, it will be in this folder, named according to its internal ID number. You can cross-reference these ID numbers by reading the .scrivx file, if necessary.
- ProjectNotes: a folder which stores any project notes you have created. Notes will be stored in RTF files, and the index.xml file defines any custom properties they may have.
- search.indexes: this file is used to optimise the search speed of Scrivener. It contains a complete plain-text copy of nearly every string in the project.

- user.lock: will only appear while the project is open (or in the unfortunate case of a crash wherein the software never got a chance to remove the file). This stores a key that identifies your computer. If you try to open the project a second time on another computer, you will be warned that it appears to be open already.
- version.txt: a simple marker for letting Scrivener know what version of the project format this is, so that it can remain compatible with future versions of the software.

E.2 The Settings Folder

This contains all of the settings which are specific to the project. Scriptwriting, compile, and overall general UI settings such as whether you had the interface split, and what they were viewing, all the way down to whether you're tinting icons with the label colour and so on. If the UI has become damaged for some reason and the project refuses to load, or loads in a malfunctioned state, it can sometimes help to reset the UI by removing the configuration file for it. This will return your project to a factory default state in terms of visual and workflow settings *only*. It will not harm any data. Make sure the project is closed, first.

The easiest way to reset the UI, if you can get the project open, is to hold down the Option key and use the **File ► Close Project and Clear Interface Settings** menu command. However if you cannot open the project, then go into this folder and remove the **ui.plist** file by hand.

E.3 The Snapshots Folder

The snapshots folder is where all of your old snapshots are stored. A new sub-folder will be created within it for each item in the binder that has snapshots. The folder contains an XML file with some information about the snapshot (such as its descriptive name and time stamp), as well as any link information that existed when the snapshot was taken. The snapshots themselves are standard RTF files, and so can be accessed in an emergency if necessary.

The presence of these folders are the one thing that Scrivener uses to determine if an item in the binder has snapshots. If you delete the folder, it will no longer have snapshots. Thus if you are intending to distribute the .scriv file for whatever reason, and wish to wipe out all of the old working data, it is safe to delete these folders on a one-by-one basis, or indeed the entire Snapshots folder itself to totally clean out the project of all old versions. You will probably only ever want to do that on a copy of the original project.

Appendix F

Legal

Scrivener and this manual are copyright © 2006–2015 by Literature & Latte LLC. All rights reserved. Literature & Latte are not affiliated with, or endorsed by the following companies and their products or services:

- Microsoft Office (and all products therein); Microsoft Windows
- Apple iWork, iCloud, Mac OS X and all utilities mentioned therein
- Adobe Acrobat Pro; Adobe Reader
- Amazon; Kindle, and KindleGen tool
- Aspose
- Bookends
- Calibre
- CeltX
- Courier Prime
- DenVog Index Card for iPad
- Dropbox
- EndNote
- Final Draft
- Fountain
- Freemind
- Goodle
- Hog Bay Software; PlainText; WriteRoom

- Mariner Montage
- Movie Magic Screenwriter
- Nisus Writer Pro
- OpenOffice.org (and derivatives NeoOffice & LibreOffice)
- Papyrus
- RedleX Mellel
- Simperium Simplenote
- Sony Reader
- SpiderOak
- Twitter
- WordNet

All products above and otherwise mentioned in this text are trademarks of their respective owners.

Any person names used in illustrations or screenshots are fictional and used fictitiously. Any resemblance to actual events, locales, or persons, living or dead, is coincidental.

Literature & Latte take no responsibility for any text lost through synchronisation. As with all synchronisation processes, you should ensure you back up regularly and double-check that the right documents are being synchronised.

Visit our web site for a copy of the licence agreement[1]. By using this software you are agreeing to be bound by the terms of this licence.

[1] http://www.literatureandlatte.com/licence.php